

ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

LUIS FERNANDES VAZ

**DOCUMENTAÇÃO DE TAREFAS EM SOFTWARE CROWDSOURCING:
UM ESTUDO EMPÍRICO SOBRE A PLATAFORMA TOPCODER**

Porto Alegre
2018

PÓS-GRADUAÇÃO - *STRICTO SENSU*



Pontifícia Universidade Católica
do Rio Grande do Sul

Ficha Catalográfica

V393d Vaz, Luis Fernandes

Documentação de Tarefas em Software Crowdsourcing : Um Estudo Empírico sobre a Plataforma TopCoder / Luis Fernandes Vaz . – 2018.

182.

Dissertação (Mestrado) – Programa de Pós-Graduação em Ciência da Computação, PUCRS.

Orientadora: Profa. Dra. Sabrina Marczak.

1. Software Crowdsourcing. 2. Documentação de Tarefas. 3. Engenharia de Requisitos. 4. TopCoder. 5. Estudo Empírico. I. Marczak, Sabrina.
II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da PUCRS
com os dados fornecidos pelo(a) autor(a).
Bibliotecária responsável: Salete Maria Sartori CRB-10/1363

LUIS FERNANDES VAZ

**DOCUMENTAÇÃO DE TAREFAS EM SOFTWARE CROWDSOURCING:
UM ESTUDO EMPÍRICO SOBRE A PLATAFORMA TOPCODER**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação, Faculdade de Informática da Pontifícia Universidade Católica do Rio Grande do Sul.

Aprovado em 27 de Março de 2018.

BANCA EXAMINADORA:

Prof. Dr. Rafael Prikladnicki (PPGCC/PUCRS)

Prof. Dr. Cleidson de Souza (PPGCC/UFPa)

Prof^a. Dr^a. Sabrina dos Santos Marczak (PPGCC/PUCRS - Orientadora)

Dedico este trabalho a meus pais,
Domingos e Reli (*in memoriam*) e
à minha esposa Daniela.

AGRADECIMENTOS

Agradeço à minha família, em especial a minha esposa Daniela, aos meus pais, Domingos e Reli (*in memoriam*), enfim a todos que comigo assumiram este desafio, me dando a sustentação necessária para concluir esta etapa importante da minha vida.

Aos amigos, que compreenderam a escolha pela realização do Mestrado e, portanto, os momentos de ausência devido à pesquisa e escrita desta dissertação.

À Professora Dra. Sabrina Marczak, minha orientadora, pela oportunidade de aprendizado, pelos momentos de orientação, por sua amizade, dedicação, profissionalismo e apoio para a realização deste estudo.

Ao professor Dr. Igor Steinmacher, pelo apoio e contribuições em diversos momentos durante a realização desta pesquisa.

Aos professores Dr. Rafael Prikladnicki, por ter acompanhado e contribuído em todas as etapas do mestrado inclusive aceitando participar da Banca de Defesa desta dissertação, e Dr. Cleidson de Souza, que aceitou gentilmente participar da Banca de Defesa desta dissertação.

Aos professores e aos colegas do Programa de Pós-Graduação em Ciência da Computação da PUCRS, e principalmente aos colegas do Grupo MuNDDoS pelos ensinamentos e pelo convívio enriquecedor.

À empresa Dell Inc. por ter proporcionado a bolsa de estudos para meu mestrado sob a Lei de Informática (Lei 8.248/91).

Ao Departamento de Informática do Tribunal de Justiça do Rio Grande do Sul, representado pelos diretores da área de sistemas Débora Pritsch, Clairton Buligon e Carolina Mobus, pelo apoio em todo o período de realização do mestrado. Também agradeço à compreensão dos meus colegas Luciano e Rafael, pelos momentos em que necessitei me ausentar das minhas atividades profissionais para dar continuidade ao mestrado.

Enfim, agradeço a todas as pessoas que de alguma maneira contribuíram para o desenvolvimento desta pesquisa.

DOCUMENTAÇÃO DE TAREFAS EM SOFTWARE CROWDSOURCING: UM ESTUDO EMPÍRICO SOBRE A PLATAFORMA TOPCODER

RESUMO

A presente pesquisa teve como objetivo investigar a documentação das tarefas disponibilizadas na plataforma TopCoder e os elementos que devem ser considerados na documentação de uma tarefa em *Software Crowdsourcing*. Esta investigação torna-se relevante na medida em que a Tarefa é o elemento fundamental de ligação entre os demais elementos do modelo de *Software Crowdsourcing* (Contratante, Plataforma e Multidão). É a Tarefa que expressa a necessidade do Contratante para os membros da multidão. Assim, para o desenvolvimento desta investigação foi adotada a abordagem qualitativa, por meio de um Estudo de Caso com novatos em *Software Crowdsourcing* e de um Estudo de Campo, com profissionais da indústria. Para a análise e interpretação dos dados foi aplicada a técnica de Análise de Conteúdo. Como resultado desta pesquisa, constatou-se que no Estudo de Caso a documentação da tarefa teve um papel secundário quando os participantes selecionavam as tarefas. Entretanto, o papel da clareza da documentação surge com maior força durante a execução da tarefa, uma vez que é neste momento que deve ser decodificada a instrução da documentação a fim de realizar efetivamente a tarefa e submetê-la à plataforma. Para os participantes do Estudo de Campo, os elementos mais relevantes referentes à documentação das tarefas foram a clareza na descrição da tarefa e o conhecimento sobre o assunto tratado pela tarefa. A partir dos resultados obtidos é proposto um modelo de documentação de tarefa a ser utilizado na plataforma TopCoder. Acredita-se que com o mapeamento dos elementos identificados na pesquisa e a proposta de um modelo de documentação para a tarefa será possível aprimorar a descrição das tarefas e conseqüentemente as entregas realizadas pelos membros da multidão.

Palavras Chave: *Software Crowdsourcing*, Documentação de Tarefas, Engenharia de Requisitos, TopCoder, Estudo Empírico.

SOFTWARE CROWDSOURCING TASKS DOCUMENTATION: AN EMPIRICAL STUDY OF THE TOPCODER PLATFORM

ABSTRACT

This research aimed to investigate task documentation in Software Crowdsourcing, more specifically, in the TopCoder platform. It also aimed to identify the elements that should be considered in the documentation of a task in this kind of software development. This research is of importance when considering that a Task is the component that links the other components of the software crowdsourcing model, which are: the Buyer, the Platform, and the Crowd. It is the task that expresses the Buyer's need to the crowd members. We followed a qualitative research approach and conducted a Case Study with newcomers in Software Crowdsourcing and a Field Study with industry professionals. Data was analyzed using the Content Analysis technique. We found that, for the Case Study novices, the documentation of the task had a secondary role in the task selection. However, the need of a clear documentation become more relevant during the development of the task given that this is the moment that the instructions within the documentation need to be decoded by the developer and turned into a solution to be later submitted to the platform. For the Field Study participants, the most relevant elements related to the documentation of a task were how clear the description of a task is and their prior knowledge about the task content in order to influence its selection. Inspired on our studies' results, we propose a model for task documentation in TopCoder. We believe this model will likely aid the description of tasks in software crowdsourcing and will, as a consequence, help crowd members in their task development journey.

Keywords: Software Crowdsourcing, Task Documentation, Requirements Engineering, TopCoder, Empirical Study.

LISTA DE FIGURAS

Figura 2.1 – Fatores formadores do <i>Crowdsourcing</i>	17
Figura 2.2 – Componentes do modelo básico de <i>Crowdsourcing</i>	18
Figura 2.3 – <i>Crowdsourcing</i> e Engenharia de Software	21
Figura 2.4 – TopCoder - Processo de Desenvolvimento	24
Figura 2.5 – Modelo atual da estrutura da documentação de uma tarefa na TopCoder . . .	25
Figura 2.6 – Representação da decomposição de tarefas	26
Figura 2.7 – Processo de decomposição das tarefas e documentação	27
Figura 2.8 – Relacionamento entre tipos de requisitos	31
Figura 2.9 – Etapas do processo de desenvolvimento de requisitos	31
Figura 3.1 – Representação do Desenho da Pesquisa - Visão Geral	39
Figura 3.2 – Processo de Revisão da Literatura	42
Figura 3.3 – Representação dos modelos de pesquisa de McGrath [29]	45
Figura 3.4 – Espiral evolutiva da análise de conteúdo de Creswell [7]	47
Figura 4.1 – Consolidação da fase de avaliação de artigos da revisão da literatura	51
Figura 4.2 – Modelo evolutivo de Micro Multidões (MM) e dos requisitos obtidos	52
Figura 4.3 – Etapas do processo de seleção e documentação de requisitos [12]	53
Figura 4.4 – Fluxo de processamento e validação de tarefas proposto por FU [9]	55
Figura 5.1 – Processo de avaliação das informações (Questionários e Diário de Bordo) . . .	68
Figura 5.2 – Qualificação da documentação no Estudo de Caso - Ciclo 1 e 2	71
Figura 5.3 – Fragmento da avaliação da documentação da tarefa no Estudo de Caso (P1) . . .	74
Figura 5.4 – Fragmento da avaliação da documentação da tarefa no Estudo de Caso (P11) . . .	74
Figura 6.1 – Fragmentos de tarefas finalizadas na plataforma TopCoder	88
Figura 7.1 – Comparativo da descrição de tarefas na plataforma TopCoder	96
Figura 7.2 – Proposta de documentação de uma tarefa para a plataforma TopCoder	99

LISTA DE TABELAS

Tabela 1.1 – Modelo base para documentação de tarefas proposto por Shao [33]	13
Tabela 2.1 – Conceitos de <i>Crowdsourcing</i>	18
Tabela 2.2 – Plataformas comerciais de <i>Crowdsourcing</i>	22
Tabela 2.3 – Tipos de requisitos	30
Tabela 2.4 – Boas práticas para a engenharia de requisitos	32
Tabela 2.5 – Critérios individuais de excelência para documentar requisitos	33
Tabela 2.6 – Critérios de excelência para documentar requisitos em conjunto	33
Tabela 2.7 – Características de qualidade de um Requisito - Visão BABOK 2	34
Tabela 2.8 – Características de qualidade de um Requisito - Visão BABOK 3	35
Tabela 4.1 – Critérios de avaliação dos artigos selecionados	52
Tabela 4.2 – Comparativo entre a ER Tradicional e a ER <i>Crowd-Based</i>	54
Tabela 4.3 – Modelo base para documentação de tarefas proposto por Shao [33]	56
Tabela 5.1 – Resumo dos dados de perfil dos participantes do Estudo de Caso	59
Tabela 5.2 – Plano de Ação do Estudo de Caso para o Ciclo 1 e Ciclo 2	62
Tabela 5.3 – Listagem com os critérios de classificação da documentação da tarefa - Ciclo 2	73
Tabela 5.4 – Classificação da documentação da tarefa no Estudo de Caso - Ciclo 1 e 2 . . .	73
Tabela 5.5 – Classificação das sugestões do Estudo de Caso	75
Tabela 5.6 – Experiência em linguagens avaliadas por categorias no Estudo de Caso	79
Tabela 6.1 – Experiência em TI dos participantes do Estudo de Campo	81
Tabela 6.2 – Plano de Ação do Estudo de Campo	81
Tabela 6.3 – Critérios de Seleção/Descarte obtidos no Estudo de Campo	86
Tabela 6.4 – Seleção das tarefas pelos participantes do Estudo de Campo	87
Tabela 6.5 – Análise da seleção das tarefas pelos participantes no Estudo de Campo	89
Tabela 6.6 – Classificação da documentação pelos participantes do Estudo de Campo	89
Tabela 6.7 – Categorização da documentação pelos participantes do Estudo de Campo . . .	91
Tabela 6.8 – Critérios de avaliação da documentação obtidos no Estudo de Campo	91
Tabela 6.9 – Sugestões de melhorias para a documentação das tarefas - Estudo de Campo	93
Tabela 7.1 – Classificação dos elementos utilizados no modelo de documentação	100
Tabela 7.2 – Critérios de classificação de requisitos aplicados ao modelo de documentação	105

LISTA DE SIGLAS

BABOK – *A Guide to the Business Analysis Body of Knowledge*

IBM – *International Business Machines*

MUNDDOS – *A Research Group on Global Software Development*

PUCRS – Pontifícia Universidade Católica do Rio Grande do Sul

TI – Tecnologia da Informação

SUMÁRIO

1	INTRODUÇÃO	12
1.1	CONTEXTO, MOTIVAÇÃO E PROBLEMÁTICA	12
1.2	OBJETIVOS GERAL E ESPECÍFICOS	14
1.3	CONTRIBUIÇÕES	14
1.4	ESTRUTURA DA DISSERTAÇÃO	15
2	REFERENCIAL TEÓRICO	16
2.1	CROWDSOURCING	16
2.2	SOFTWARE CROWDSOURCING	20
2.2.1	PLATAFORMAS DE CROWDSOURCING	20
2.2.2	PLATAFORMA TOPCODER	22
2.3	TAREFAS EM SOFTWARE CROWDSOURCING	24
2.4	ENGENHARIA DE REQUISITOS E QUALIDADE DA DOCUMENTAÇÃO	28
3	METODOLOGIA DE PESQUISA	37
3.1	MÉTODO DE PESQUISA	37
3.2	DESENHO DA PESQUISA	39
3.3	REVISÃO DA LITERATURA	40
3.4	ESTUDO DE CASO	42
3.5	ESTUDO DE CAMPO	43
3.6	ANÁLISE DE CONTEÚDO	46
4	REVISÃO DA LITERATURA	49
4.1	PROTOCOLO DA REVISÃO DA LITERATURA	49
4.1.1	FASE I – PLANEJAMENTO	49
4.1.2	FASE II – EXECUÇÃO	50
4.1.3	FASE III – RESULTADOS	51
4.2	CONSIDERAÇÕES SOBRE A REVISÃO DA LITERATURA	57
5	ESTUDO DE CASO	58
5.1	SUJEITOS DA PESQUISA	58
5.2	PLANEJAMENTO E EXECUÇÃO	61
5.3	COLETA DE DADOS	66
5.4	RESULTADOS DO ESTUDO DE CASO	68

5.5	DISCUSSÃO SOBRE O ESTUDO DE CASO	78
5.6	LIMITAÇÕES DO ESTUDO DE CASO	79
6	ESTUDO DE CAMPO	80
6.1	SUJEITOS DA PESQUISA	80
6.2	PLANEJAMENTO E EXECUÇÃO	80
6.3	COLETA DE DADOS	82
6.4	RESULTADOS DO ESTUDO DE CAMPO	85
6.5	DISCUSSÃO SOBRE O ESTUDO DE CAMPO	92
6.6	LIMITAÇÕES DO ESTUDO DE CAMPO	93
7	MODELO PARA DOCUMENTAÇÃO DE TAREFAS NA TOPCODER	95
7.1	CONTEXTO	95
7.2	MODELO DE DOCUMENTAÇÃO	97
8	CONSIDERAÇÕES FINAIS	108
8.1	LIMITAÇÕES	108
8.2	TRABALHOS FUTUROS	109
	REFERÊNCIAS	110
	APÊNDICE A – Estudo de Caso: Formulário Perfil do Participante	114
	APÊNDICE B – Estudo de Caso: Formulário Seleção da Tarefa (Ciclo 1)	119
	APÊNDICE C – Estudo de Caso: Diário de Bordo	121
	APÊNDICE D – Estudo de Caso: Formulário Avaliação da Tarefa (Ciclo 1)	122
	APÊNDICE E – Estudo de Caso: Formulário Seleção da Tarefa (Ciclo 2)	127
	APÊNDICE F – Estudo de Caso: Formulário Avaliação da Tarefa (Ciclo 2)	129
	APÊNDICE G – Estudo de Campo: Termo de Confidencialidade	135
	APÊNDICE H – Estudo de Campo: Formulário Avaliação da Tarefa com Profissionais	136
	APÊNDICE I – Estudo de Caso: Sugestões de Melhorias Ciclo 1 e Ciclo 2	176

1. INTRODUÇÃO

Este capítulo tem como objetivo apresentar as motivações para a realização desta pesquisa. Também são descritas as atividades realizadas no decorrer deste trabalho e os resultados obtidos na pesquisa sobre a documentação de tarefas disponibilizadas para os membros da multidão no modelo de *Software Crowdsourcing*.

1.1 Contexto, Motivação e Problemática

Para acompanhar as mudanças que ocorreram nos últimos anos advindas da globalização e que envolvem, dentre outras áreas, os meios de produção, o processo de desenvolvimento de software também precisou se adaptar. Surgem diferentes modelos que buscam um modo eficiente e diversificado para produzir software, dentre eles o modelo de desenvolvimento globalizado. Com a queda significativa de restrições econômicas mundiais, este modelo de desenvolvimento globalizado torna-se mais consistente e presente, e assim surge o modelo de *Global Software Engineering* – GSE, em que o software é produzido de forma distribuída com os parceiros definidos no projeto. A partir deste cenário, desponta o *Software Crowdsourcing*, que visa ampliar ainda mais a capacidade de produzir software com uma base de fornecedores muito maior e que não necessita possuir uma relação direta com o projeto original, fato este que não ocorre com o GSE.

Quando se avalia o modelo de *Crowdsourcing* sob o aspecto da execução das tarefas é frequente a indicação de que as micro tarefas disponibilizadas para os membros da multidão são de simples entendimento e execução (*Human Intelligence Tasks*). Neste sentido, qualquer pessoa interessada poderia realizar este tipo de tarefa, uma vez que não é necessário conhecimento especializado [26]. No entanto, esta premissa não se torna factível quando se verifica o modelo de *Software Crowdsourcing*, em que as tarefas possuem um alto grau de especialização, logo o membro da multidão deve possuir conhecimento e experiência que o habilite a realizar este tipo de atividade [37]. Neste contexto, em que as tarefas propostas para o modelo de *Software Crowdsourcing* exigem um determinado conhecimento e possuem um alto grau de dependência com outras tarefas, torna-se ainda mais relevante que o detalhamento do que deve ser executado seja claro e objetivo. Isso significa que a documentação das tarefas deve ser capaz de representar exatamente o que o Contratante espera receber a fim de evitar retrabalho e *bugs* na solução da tarefa submetida.

A partir deste contexto, pode-se salientar a relevância da documentação das tarefas, uma vez que ela deve ser capaz de fornecer o entendimento necessário para que os membros da multidão consigam realizar a tarefa. Por isso, reforça-se que a documentação da tarefa é um fator essencial para que a mesma seja selecionada e realizada com sucesso. Destaca-se ainda, que a clara identificação do que deve ser realizado é um aspecto considerado pelos membros da multidão quando selecionam uma tarefa conforme pode ser constatado nesta pesquisa.

Tabela 1.1 – Modelo base para documentação de tarefas proposto por Shao [33]

Característica	Descrição	Tipo
Data Limite	Quanto tempo os desenvolvedores possuem para realizar a tarefa	Numérico
Recompensa	Quanto irá ganhar um desenvolvedor depois que ele for considerado o vencedor (será um intervalo de valores, média de valores)	Numérico
Validações Intermediárias	Quantas verificações de uma tarefa o contratante solicitará em média	Numérico
Plataforma	Plataformas que os desenvolvedores deverão utilizar, por exemplo: "Android", "Web"	Categórico
Linguagem	Linguagens que os desenvolvedores deverão utilizar, por exemplo: "Java", "Python"	Categórico
Tipo	Qual é o tipo da tarefa, por exemplo "Corrigir erros", "Desenvolver software"	Categórico
Descrição	Descrições detalhadas definidas pelo contratante, incluindo o título	Documento

Fonte: Adaptado de [33]

Neste cenário, em que a documentação é o ponto de partida para que os membros da multidão avaliem as tarefas que pretendem realizar, torna-se fundamental a correta fragmentação da mesma. Inclusive esta questão pôde ser observada durante a preparação do Estudo de Caso, quando verificou-se um caso real em que um mesmo documento de requisitos foi disponibilizado em duas tarefas na plataforma TopCoder com objetivos diferentes. Considerando que a documentação de cada tarefa deveria estar autocontida e restrita ao seu escopo, a utilização de um mesmo documento com informações abrangentes pode causar mais problemas do que auxiliar na sua execução.

Apesar da relevância da documentação da tarefa no processo de seleção e execução das tarefas no modelo de *Software Crowdsourcing*, este tema é pouco explorado na literatura científica quando se busca informações referentes ao seu impacto no processo de desenvolvimento de software. Este fato pôde ser constatado na revisão da literatura sobre o tema realizada por Vaz [38], foram identificados poucos artigos relatando pesquisas que envolvam a documentação das tarefas ou seus impactos na execução das mesmas. A maior recorrência de artigos trata da identificação de requisitos a partir da multidão, ou seja, a multidão atuando como *stakeholders*. Dentre os resultado da revisão da literatura, cabe destacar o material proposto por Shao [33], que descreve um *framework* ou modelo para recomendação de participantes que se baseia em uma estrutura de documentação simplificada. Este modelo pode ser observado na Tabela 4.3. Esta proposta identifica um conjunto restrito de elementos que permitem auxiliar na identificação dos participantes capazes de realizar a tarefa. Entretanto, estes elementos não abordam a documentação da tarefa de forma completa ou mesmo como a mesma foi registrada ou sua estrutura.

Considerando este cenário, em que a documentação da tarefa é um fator crucial para o processo de trabalho em *Software Crowdsourcing*, é que esta pesquisa se enquadra. Assim, a partir desta investigação pretendeu-se obter as percepções dos membros da multidão em relação à documentação das tarefas, além de propor um modelo que traga maior detalhamento e clareza a esta documentação.

1.2 Objetivos Geral e Específicos

O objetivo geral deste estudo foi investigar a documentação das tarefas disponibilizadas na plataforma TopCoder e os elementos que devem ser considerados na documentação de uma tarefa em *Software Crowdsourcing*. Para que fosse possível atingir o objetivo geral, foram definidos os seguintes objetivos específicos:

- **Objetivo 1:** Caracterizar como os membros da multidão percebem a documentação das tarefas disponíveis na plataforma TopCoder.
- **Objetivo 2:** Identificar quais são os elementos que devem compor a descrição de uma tarefa.

1.3 Contribuições

Ao realizar esta pesquisa foi possível identificar como contribuições dois aspectos, o primeiro está relacionado diretamente à geração do registro científico da pesquisa, já o segundo faz referência a uma proposta de melhoria na documentação de tarefas em *Software Crowdsourcing*. Estes pontos são detalhados a seguir:

- Como primeiro item pode-se considerar o efetivo registro da pesquisa. Esta contribuição torna-se relevante tendo em vista a quantidade reduzida de produção científica que trata do tema abordado no trabalho, a documentação da tarefa. Conforme identificado na revisão da literatura realizada pesquisador, o conjunto de artigos obtidos sobre o tema de *Software Crowdsourcing* trata majoritariamente da atuação dos membros da multidão como *stakeholders* no processo de identificação de requisitos. Neste sentido, esta pesquisa apresenta uma visão diferenciada em relação à documentação, ou seja, descreve como a documentação influencia no comportamento destes participantes ao interagirem com a plataforma. Logo, o registro desta pesquisa amplia o conhecimento científico, viabilizando, assim, que novas pesquisas possam utilizar como base os conceitos e informações aqui identificadas.
- Como segunda contribuição desta pesquisa destaca-se a proposta de um modelo para a estrutura das informações registradas em uma tarefa da plataforma TopCoder, isto é, a documentação da tarefa. Com esta alteração da estrutura pretende-se aprimorar as informações fornecidas aos membros da multidão, gerando uma documentação padronizada, clara e objetiva da tarefa. Com esta proposta acredita-se que os participantes poderão ser mais efetivos na resolução das tarefas selecionadas.

1.4 Estrutura da Dissertação

Este trabalho está organizado em sete capítulos. No Capítulo 2 são apresentados os principais conceitos relacionados ao tema de pesquisa, que são: *Crowdsourcing*, *Software Crowdsourcing*, Plataformas de *Crowdsourcing*, Plataforma TopCoder, tarefas em *Software Crowdsourcing* e Engenharia de Requisitos.

O Capítulo 3 descreve a metodologia utilizada para o desenvolvimento deste estudo, indicando as etapas percorridas e relatando a escolha pelos métodos de pesquisa aplicados nessa investigação.

No Capítulo 4 é apresentada de forma resumida a Revisão da Literatura realizada para esta pesquisa, onde são descritos o protocolo de pesquisa e os resultados obtidos. A versão completa desta revisão da literatura pode ser obtido no trabalho de Vaz [38].

O Capítulo 5 apresenta de forma detalhada o planejamento, as ações e os resultados obtidos a partir do Estudo de Caso realizado. São descritos o cenário em que o Estudo de Caso foi desenvolvido, os sujeitos da pesquisa, os instrumentos e procedimentos utilizados para a coleta de dados, bem como todas as etapas e os resultados encontrados.

O Capítulo 6 trata de descrever em detalhes cada etapa executada para a realização do Estudo de Campo. São apresentados o cenário em que o estudo foi desenvolvido, os sujeitos da pesquisa, os instrumentos e procedimentos utilizados para a coleta de dados e os resultados obtidos.

O Capítulo 7 apresenta um modelo de documentação das tarefas na plataforma TopCoder que tem por objetivo melhorar a organização e viabilizar um melhor entendimento das tarefas. Esta proposta foi construída com base nos achados obtidos através do Estudo de Caso e do Estudo de Campo e no material científico relativo ao tema de pesquisa.

No Capítulo 8 são apresentadas as considerações finais, as limitações identificadas nesta pesquisa, assim como são sugeridos trabalhos futuros, decorrentes do conhecimento obtido através dessa investigação.

2. REFERENCIAL TEÓRICO

Nesta seção do trabalho são apresentados os principais conceitos que norteiam o desenvolvimento desta pesquisa, que são: *Crowdsourcing*, *Software Crowdsourcing*, Tarefas de Desenvolvimento de Software e Engenharia de Requisitos.

2.1 Crowdsourcing

De forma sucinta, *Crowdsourcing* pode ser descrito como um modelo de negócio alternativo à terceirização que utiliza as capacidades humanas e da computação combinadas para resolver um determinado problema. Segundo Howe [16], o *Crowdsourcing* originou-se a partir do movimento de *Open Source Software*, evidenciando que um grupo ou comunidade motivada e com um objetivo comum é capaz de criar um produto de alta qualidade.

Tendo em vista o interesse cada vez maior das empresas pelo *Crowdsourcing*, faz-se necessário destacar os principais fatores que impulsionaram a formação do *Crowdsourcing*. Conforme demonstrado na Figura 2.1, o *Crowdsourcing* surge da intersecção de três elementos, são eles: *Outsourcing* (ou Terceirização), *Crowd* (ou Multidão) e *Social Web* (ou tecnologias Internet). Esses elementos são descritos a seguir [32]:

- "*Outsourcing*" ou terceirização:

Conforme a proposta de Howe [16], o *crowdsourcing* representa um tipo de terceirização emergente em pequena escala, uma vez que trata da aquisição de bem ou serviço de um prestador externo. Esta comparação entre o *crowdsourcing* e a terceirização ocorre porque os dois modelos compartilham um objetivo básico em comum, que é alcançar suas metas de negócio através de uma entidade externa capaz de solucionar um determinado problema.

- "*Crowd*" ou multidão:

Apesar de existir uma relação entre *crowdsourcing* e a terceirização, visto que ambos alcançam seus objetivos através de uma entidade externa, o mesmo não ocorre quando se trata dos entes que realizam estas atividades (prestadores de serviços ou fornecedores). Isso porque a terceirização possui a identificação clara e explícita das partes envolvidas no processo e o que elas devem realizar, já o modelo de *crowdsourcing* não possui essa identificação clara. No *crowdsourcing* existe uma multidão *on-line* que pode desempenhar as atividades solicitadas. Essa multidão desempenha o importante papel de provedor de serviço, com a vantagem de que a partir dela é possível dispor de um número ilimitado de potenciais profissionais.

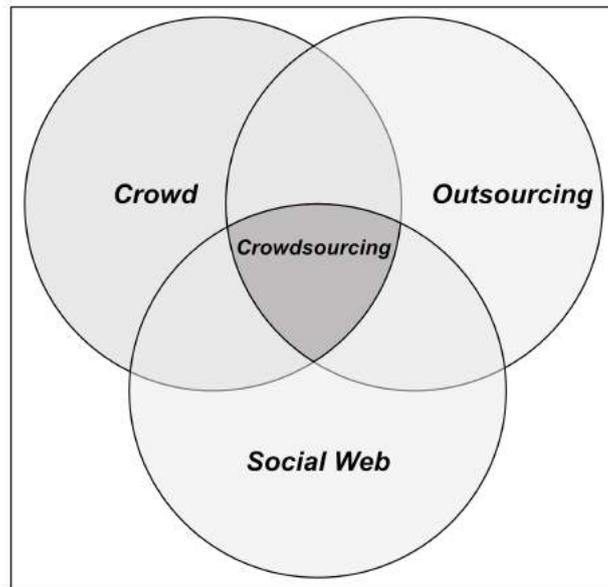


Figura 2.1 – Fatores formadores do *Crowdsourcing*

Fonte: Adaptado de [32]

- "*Social Web*" ou tecnologias Internet:

Viabilizam a operacionalização do *crowdsourcing* em larga escala uma vez que a Web 2.0 apresenta um potente conjunto de ferramentas que simplificam a comunicação e o compartilhamento de conteúdo. Devido a sua grande abrangência, no modelo de *crowdsourcing* estas tecnologias são utilizadas para encontrar a potencial multidão de trabalhadores, além de negociar contratos e acompanhar o progresso do trabalho em tempo real. Desse modo, qualquer empresa pode aproveitar as tecnologias para terceirizar uma grande variedade de tarefas para uma multidão *on-line*.

A partir do entendimento sobre os elementos que originaram o *Crowdsourcing*, são destacadas na Tabela 2.1 algumas definições comumente utilizadas sobre este tema, apontadas por Lebraty [23]. Convém salientar que o conceito de *Crowdsourcing* foi apresentado pela primeira vez em 2006, por Howe, em um *blog* e posteriormente em um artigo da revista *Wired*¹. Como este é um conceito relativamente novo e aberto, ainda não existe um consenso sobre a sua definição [14].

Assim, baseado nas definições apresentadas na Tabela 2.1 é possível constatar que existem alguns elementos similares entre elas, conforme indicado a seguir: (i) a existência de um problema estabelecido ao qual se procura obter uma solução; (ii) a realização de uma chamada aberta para que os interessados saibam da existência da atividade e (iii) a multidão, capaz e disposta a realizar uma tarefa.

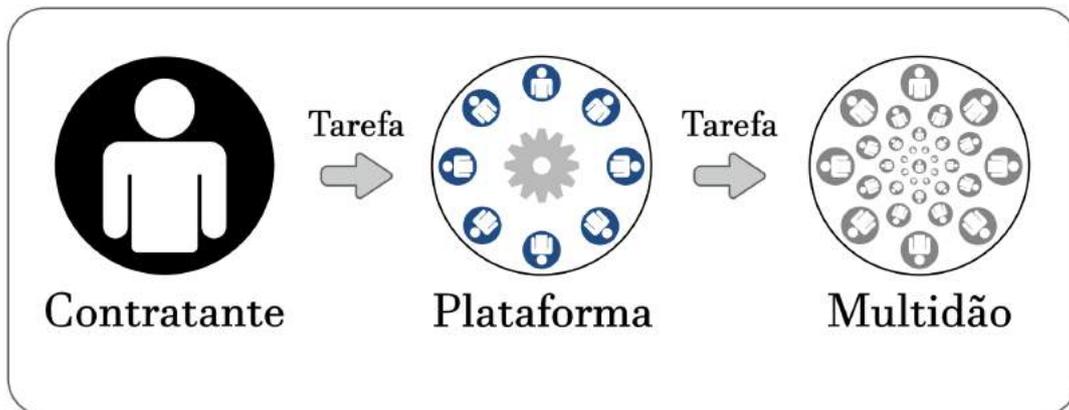
Fundamentado nestes elementos de consenso foi proposto um modelo base para a operação do *Crowdsourcing*. Este modelo é dividido em três segmentos conforme exibido na Figura 2.2. Em uma ponta deste fluxo encontra-se o Contratante (ou "*Buyer*"), que é o componente que propõe a

¹*Wired* é uma revista norte-americana que aborda questões sobre tecnologia, ciência, negócios, etc.

Tabela 2.1 – Conceitos de *Crowdsourcing*

Definição	Autor
<i>Crowdsourcing</i> é o ato de terceirizar para um grupo indefinido e grande de pessoas através de uma chamada aberta.	Jeff Howe [HOW08]
<i>Crowdsourcing</i> é a canalização do desejo dos especialistas em resolver um problema e, em seguida, compartilhar livremente a resposta com todos.	Van Ess, Henk. [VAN10]
Diz-se que um sistema é <i>Crowdsourced</i> quando ele envolve uma multidão de pessoas para ajudar a resolver um problema definido pelos proprietários do sistema, e se, ao fazê-lo, ele aborda os quatro desafios fundamentais: <ol style="list-style-type: none"> 1. Como recrutar e reter os usuários? 2. Que contribuições os usuários podem fazer? 3. Como combinar contribuições dos usuários para resolver o problema definido? 4. Como avaliar os usuários e suas contribuições? 	Doan, Anhai; Ramakrishnan, Raghu; Halevy, AlonY. [DOA11]
<i>Crowdsourcing</i> é um neologismo para o ato de pegar tarefas tradicionalmente realizadas por um empregado ou contratado e terceirizá-las para um grupo indefinido, geralmente grande, de pessoas ou uma comunidade na forma de uma chamada aberta.	Bell Dan [BEL09]

Fonte: Adaptado de [23] (p. 16-17).

Figura 2.2 – Componentes do modelo básico de *Crowdsourcing*

Fonte: Adaptado de [26]

tarefa a ser realizada. Na outra extremidade do fluxo há a Multidão (ou "*Crowd*"), que é composta por todos os entes que estão dispostos a realizar as tarefas (técnicos e não técnicos) conforme o tipo de atividade. Entre estes dois entes há a Plataforma (ou "*Crowdsourcing Platform*"), que é o ente que faz a intermediação entre os dois membros do processo: o contratante e multidão, ou seja, representa a relação entre os entes envolvidos no *crowdsourcing* [30, 26]. Entre todos estes elementos existe a Tarefa (ou "*Task*"), que representa as atividades que são propostas pelo contratante. As tarefas são decompostas e gerenciadas pela plataforma, sendo realizadas pela multidão [14].

Com base neste conceito, em que uma multidão participa de forma cooperativa para atingir um determinado objetivo, foram identificadas diferentes abordagens no que diz respeito à forma de participação da multidão. Nesse sentido, nota-se que a natureza do tipo de contribuição pode influenciar de forma significativa na maneira como a multidão irá atuar [16, 4]. Desse modo é possível subdividir a forma de participação da multidão nas seguintes categorias:

- A inteligência coletiva ou sabedoria da multidão (*Collective intelligence or crowd wisdom*)

Esta categoria tem como premissa que o conhecimento da multidão é maior do que o conhecimento dos indivíduos. Logo, o *Crowdsourcing* surge como uma forma de propiciar a esta multidão as condições necessárias para que elas possam demonstrar estes conhecimentos. Este tipo de atuação é bem ampla e pode ser aplicada a diferentes contextos.

- Criatividade da multidão (*Crowd creation*)

Esta categoria visa obter da multidão o seu poder e energia criativa, dando ênfase em ações que envolvem tarefas mais específicas, como por exemplo: a criação de comerciais ou programas para TV, dentre outras formas de comunicação.

- Votação da multidão (*Crowd voting*)

Esta categoria usa o julgamento ou opinião da multidão para organizar grandes quantidades de dados que geralmente tem como fonte o voto aberto para multidão, permitindo, assim, obter como resultado as preferências de determinados grupos.

- Financiamento colaborativo (*Crowdfunding*)

Esta categoria visa obter apoio financeiro para a realização de uma atividade e é aplicada às mais diversas finalidades, como por exemplo: apoio humanitário, fundo artístico para a realização de determinado evento, entre outras.

Da mesma forma como ainda não existe um conceito único para definir *Crowdsourcing*, a sua classificação também apresenta diferentes interpretações, como é divulgado em um estudo recente da Comissão Europeia que categoriza *Crowdsourcing* em três grupos apresentados abaixo [31] e [11]:

- *Crowdsourcing* para financiamento (*Crowdfunding*)

Iniciativa que tem como objetivo fomentar a coleta de fundos fornecidos por uma multidão de desconhecidos que se unem em torno de um ideal ou objetivo comum. Os projetos possuem as mais diversas finalidades.

- *Crowdsourcing* para o trabalho pago

Iniciativa que promove a união de um grupo (multidão) em torno de um problema proposto. A multidão compete para realizar a tarefa e assim obter retorno financeiro com a atividade.

- *Crowdsourcing* para trabalho não remunerado (voluntariado)

Iniciativa que promove o envolvimento de um grupo (multidão) na solução de um problema com o objetivo de filantropia ou reconhecimento pessoal dentro de um grupo específico.

Assim, fica claro que o modelo de *Crowdsourcing* é um modelo alternativo à terceirização, que possui uma variedade de aplicações, permitindo a sua utilização em diversos ramos de atividades, como negócios e marketing, medicina, sociologia, ciências ambientais, dentre outros [14].

2.2 Software Crowdsourcing

A área de desenvolvimento de software é inovadora e pioneira por natureza. Geralmente se dispõe a experimentar e aceitar as diversas tendências emergentes com facilidade, como ocorreu nos casos de empreendedorismo tecnológico e *outsourcing*, por exemplo, e mais recentemente com *Crowdsourcing* ou especificamente *Software Crowdsourcing*. Essa busca por inovação almejada pela indústria de software possui como fatores primordiais a redução de custos, redução do *time to market*, a gestão de talentos, maior qualidade da solução obtida, maior diversidade de soluções e a ampliação de mercados (consumidores e fornecedores) [30, 44].

Neste contexto, percebe-se que o *Crowdsourcing* preconiza um conjunto de benefícios que possui uma grande afinidade com os anseios da indústria de software, como por exemplo: a possibilidade de aumentar a capacidade produtiva através de uma força de trabalho externa e qualificada para realizar tarefas específicas [1]. Cabe destacar que um fator importante para a ampliação do uso do *Crowdsourcing*, principalmente no setor de desenvolvimento de software, é a viabilidade que a Internet de alta velocidade proporciona para a realização das tarefas a qualquer hora e em qualquer lugar [1].

Desse modo, devido à expansão do *Crowdsourcing* aplicado ao desenvolvimento de software surgiu um termo para melhor identificar as técnicas utilizadas: “*Crowdsourced Software Engineering*” que propõe reunir as técnicas de *crowdsourcing* para apoiar o desenvolvimento de software. Este conceito é abrangente, comportando um conjunto de técnicas, tais como: a elicitação de requisitos, o refinamento de casos de teste e o planejamento de projetos, arquitetura do projeto, entre outras [26].

A intersecção dos campos de conhecimento envolvendo o *Crowdsourcing* e a Engenharia de Software pode ser observada na Figura 2.3. Nota-se que surgem derivações e subáreas com focos específicos e conseqüentemente naturezas técnicas peculiares [26].

2.2.1 Plataformas de Crowdsourcing

É importante reforçar que um dos componentes mais relevantes no processo de *crowdsourcing* é a plataforma, uma vez que é ela que proporciona o canal de comunicação entre o contratante

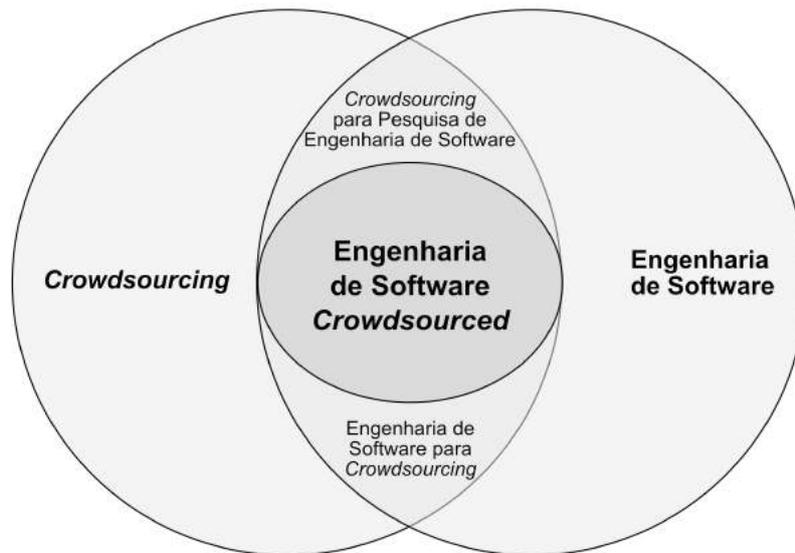


Figura 2.3 – *Crowdsourcing* e Engenharia de Software

Fonte: Adaptado de [26] (p. 5)

e a multidão, facilitando suas iterações durante o processo de realização das tarefas. Além disso, a plataforma é a “porta de entrada” para o contratante e o futuro participante.

Outro aspecto importante referente às plataformas é que devido a Engenharia de Software possuir um significativo conjunto de áreas de atuação (como por exemplo: arquitetura de software, engenharia de requisitos, *design*, teste de software, entre outros) esta variabilidade se reflete nas principais plataformas de *crowdsourcing* existentes, tendo em vista que elas podem ser segmentadas por nichos específicos, tais como [26]:

- TopCoder e GetACoder

Plataformas que suportam vários tipos de tarefas do ciclo completo de desenvolvimento de software.

- uTest e BugCrowd

Plataformas que dão suporte às atividades de teste de software e análise de segurança.

- Amazon Mechanical Turk e Freelancer

Plataformas que suportam tarefas com variado grau de complexidade. Podem ser voltadas à engenharia de software ou não.

Esta segmentação pode ser observada nas plataformas comerciais conforme exibido na Tabela 2.2, em que são apresentadas as informações consolidadas.

Tabela 2.2 – Plataformas comerciais de *Crowdsourcing*

Plataforma	Finalidade ou Fase de Desenvolvimento	Data de Início	Nº de Membros	Localização	Modelo de seleção	Recursos de aprendizagem	Sistema de Reputação
Amazon Mechanical Turk	Micro tarefas	2005	500000	Estados Unidos	Trabalho Freelance	Sim	Sim
Stack Overflow	Micro tarefas	2008	1,3 milhões	Estados Unidos	Trabalho Freelance	Sim	Sim
Freelancer	Micro tarefas	2009	18 milhões	Austrália	Trabalho Freelance	Sim	Sim
Quirky	Micro tarefas	2009	N/A	Estados Unidos	Trabalho Freelance	Sim	Sim
TopCoder	Multiplas fases	2001	1 milhão	Estados Unidos	Competição	Não	Sim
Upwork	Multiplas fases	2015	9000	Estados Unidos	Recrutamento	N/A	Sim
Corwdplat	Multiplas fases	N/A	N/A	Estados Unidos	Competição e Trabalho Freelance	Não	N/A
GetACoder	Multiplas fases	2004	N/A	Estados Unidos	Trabalho Freelance	N/A	Sim
CrowdREquire	Requisitos	2013	N/A	N/A	Competição	N/A	N/A
DesignCrowd	Design	2008	400000	Austrália	Competição e Trabalho Freelance	Não	Sim
99Design	Design	2008	300000	Estados Unidos	Competição	Não	N/A
Bountify	Desenvolvimento	N/A	N/A	N/A	N/A	N/A	N/A
uTest	Teste	2007	175000	Estados Unidos	Recrutamento	Sim	Sim
PassBrains	Teste	2011	70000	Suíça	Recrutamento	Sim	Sim
BugFinders	Teste	2010	55000	Europa	Recrutamento	N/A	N/A
Testbirds	Teste	2012	65000	Europa	Recrutamento	Não	Sim
99Tests	Teste	2010	12000	Índia	Competição	Não	Sim

Fonte: Adaptado de [44] p.114

2.2.2 Plataforma TopCoder

Dentre o conjunto de plataformas comerciais apresentadas a TopCoder se destaca como a plataforma comercial pioneira em iniciativas de *Crowdsourced Software Engineering*. O histórico desta empresa é relatado a seguir conforme descrito por Lakhani [22].

A TopCoder foi criada em 2001 quando seu fundador, Jack Hughes, decide propor uma forma inovadora de trabalho a partir de problemas identificados em suas experiências profissionais anteriores, tais como: alto custo e esforço no recrutamento de talentos, alta rotatividade dos profissionais, obsolescência na qualificação dos profissionais em virtude da evolução tecnológica do mercado, dentre outras. Com estas premissas, Hughes vislumbra a oportunidade de reduzir os custos e o esforço dos clientes no desenvolvimento de software utilizando um pool de componentes básicos já existentes ao invés de construir um sistema completo a partir do zero.

A partir desta perspectiva, a TopCoder propõe um modelo que prioriza a reutilização de componentes para solucionar o problema dos clientes e, quando necessário, realizar uma competição para a construção de novos componentes. Como forma de viabilizar esta estratégia foi necessário organizar uma comunidade capaz de atuar conforme o modelo de Hughes, isto é, identificar progra-

madores capacitados para utilizar os componentes existentes e dispostos a competir na construção de novos componentes com o objetivo de solucionar os problemas propostos pelos clientes.

Com este modelo de desenvolvimento de software definido, Hughes estrutura a empresa para trabalhar como uma “plataforma de dois lados”. De um lado a plataforma atende os clientes, que são empresas que necessitam do desenvolvimento de software customizado e que trabalham com sua equipe para especificar os desafios de programação, também definido como Contratante. Do outro lado estão os membros da comunidade, que vão competir para resolver os desafios propostos (seja de contrapartida monetária ou de habilidades), denominados de Multidão. Entre estes dois entes está a plataforma TopCoder que gerencia, define e aplica regras para clientes e membros da comunidade. Assim, a empresa necessitou estruturar seus processos para tornar viável a implantação do modelo proposto estabelecendo um conjunto de tarefas principais para cada um dos grupos de processo, conforme indicado a seguir:

1. Processos de Gestão da Plataforma:

- a Decompor os projetos dos clientes em componentes;
- b Processar as especificações do projeto do cliente;
- c Definir o tipo de competição e prêmios adequados ao projeto;
- d Definir regra consistente e imparcial para escolher os vencedores da competição;
- e Corrigir *bugs* ao final de desenvolvimento.

2. Processos de Desenvolvimento de Software:

- a Definição de Conceitos;
- b Especificação;
- c Arquitetura;
- d Produção de Componentes;
- e Montagem de Aplicações;
- f Certificação;
- g Implantação.

O processo de desenvolvimento de software proposto pela TopCoder sugere uma abordagem sistemática semelhante à *waterfall*, em que as tarefas devem seguir um fluxo determinado através da competição aberta até a sua finalização e entrega. Da mesma forma, para cada etapa da competição existe um conjunto de fases pré-determinadas que devem ser realizadas a cada iteração. Tanto o fluxo de desenvolvimento de software, como as fases da competição podem ser observadas na Figura 2.4 (b).

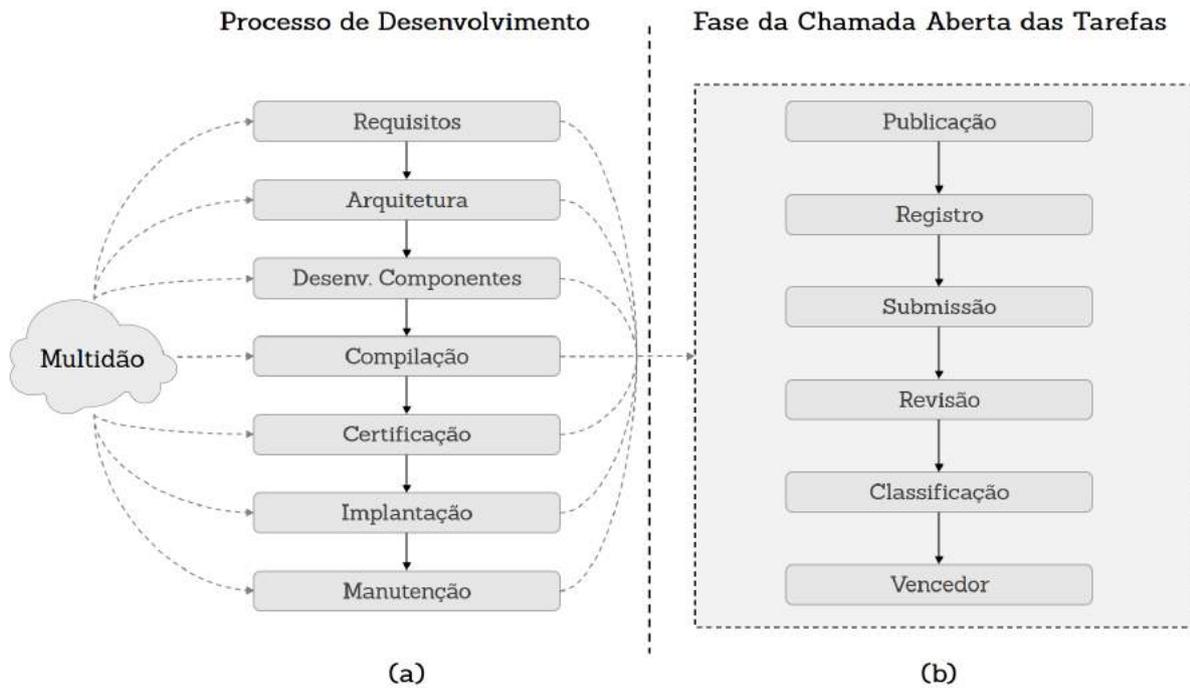


Figura 2.4 – TopCoder - Processo de Desenvolvimento

Fonte: Adaptado de [28]

Considerando que o processo de desenvolvimento de software da plataforma TopCoder segue uma abordagem *waterfall* e que o objetivo desta pesquisa trata especificamente da documentação das tarefas, cabe demonstrar como é frequentemente representada a estrutura da documentação das tarefas na plataforma, conforme apresentado na Figura 2.5. Tal estrutura é composta pelos seguintes itens: (i) Cabeçalho da tarefa, local que registra as informações básicas de identificação da mesma, tais como: nome da tarefa, prazos de cadastramento e submissão, valores relativos à premiação, dentre outros, (ii) Tópicos fixos da documentação, estão presentes em todas as tarefas avaliadas na plataforma, como por exemplo os itens: *Challenge Overview* ou *Summary*, *Payments* e *Reliability Rating and Bonus* e (iii) Tópicos complementares, são itens da documentação que são apresentados em determinados tipos de tarefas, por exemplo tarefas da categoria de *design*, em que são utilizados os seguintes itens: *Platforms*, *Technologies* e *Final Submission Guidelines*.

2.3 Tarefas em Software Crowdsourcing

De modo geral, pode-se dizer que a tarefa representa o problema que o Contratante submete à plataforma e que a multidão irá propor uma solução. Entretanto, existem diversas nuances relacionadas às tarefas que devem ser avaliadas até que estas sejam liberadas para que a multidão possa realizá-las. Conforme apresentado na Seção 2.3 (em que foi conceituado *Crowdsourcing*), dentre os elementos básicos do *crowdsourcing* há a Tarefa, que representa a atividade que foi terceirizada pelo Contratante. A Tarefa é o elemento que transita entre todos os demais: Contratante, Plataforma e Multidão. Em cada um destes entes a tarefa poderá assumir características particula-

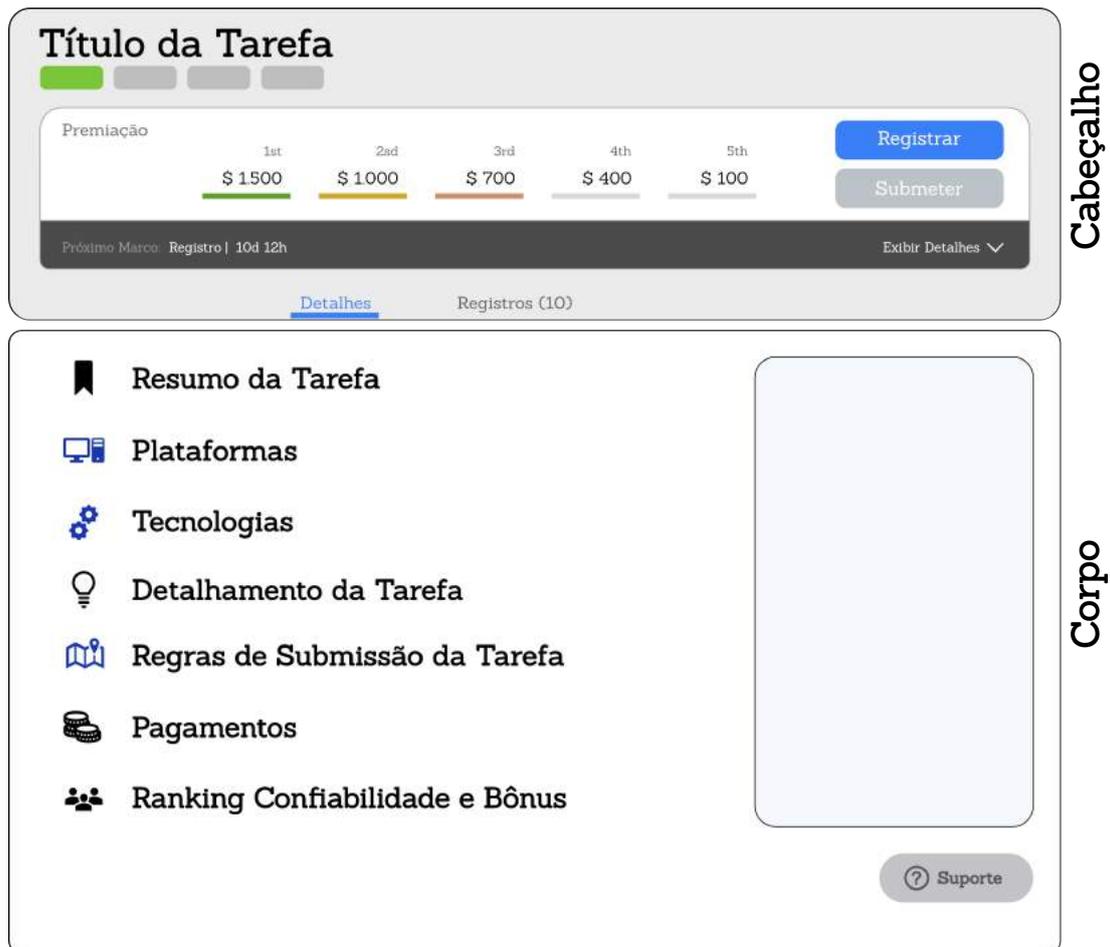


Figura 2.5 – Modelo atual da estrutura da documentação de uma tarefa na TopCoder

Fonte: Elaborado pelo autor

res relacionadas à etapa em que está envolvida. Por exemplo: quando o Contratante submete uma tarefa à plataforma, esta tarefa será avaliada para verificar a necessidade ou não de ser decomposta em tarefas menores.

Tendo em vista que a tarefa deve ser adequada para ser disponibilizada para a multidão, identifica-se uma questão crucial que envolve o *software crowdsourcing*: a Decomposição de Tarefas. De acordo com Stol et al. [37], este aspecto representa um dos grandes desafios do *software crowdsourcing*, pois é através desta ação que será possível definir como a tarefa deverá ser fragmentada em partes menores (Micro Tarefas) sem perder suas características específicas e sua interdependência com as outras partes. Este processo de decomposição de tarefas pode ser observado na Figura 2.6.

É importante salientar que no modelo puro de *Crowdsourcing* a especialização não é um fator preponderante, uma vez que as tarefas tratam com frequência de ações de baixa complexidade e baixa interdependência, comumente descritas como Human Intelligence Tasks. Nesse caso, as tarefas podem ser realizadas por participantes não especializados. Entretanto, quando se trata de *Software Crowdsourcing* há de forma recorrente tarefas de alto grau de complexidade [26], o que torna necessário que a multidão esteja capacitada para realizar este tipo de atividade [37]. Por esse

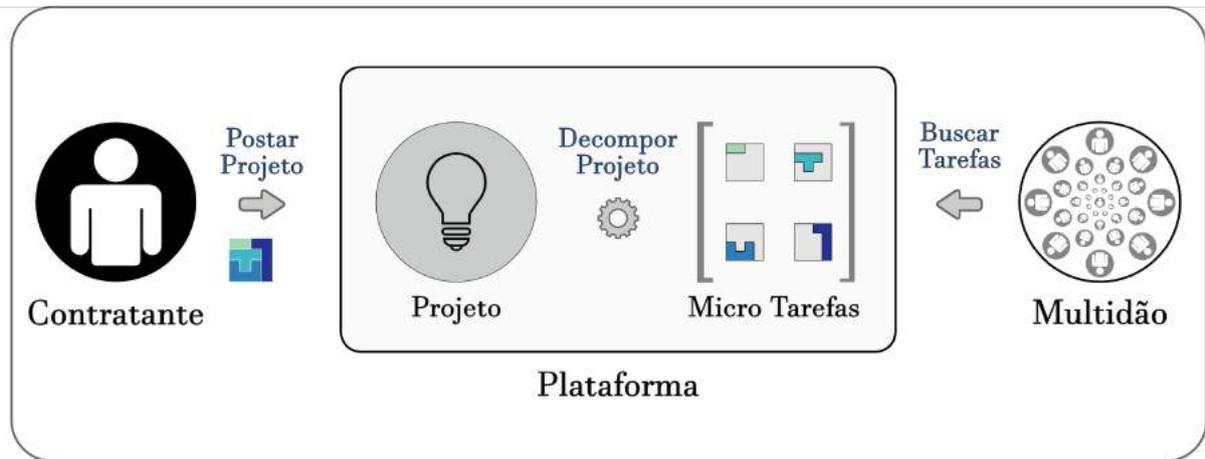


Figura 2.6 – Representação da decomposição de tarefas

Fonte: Adaptado de [37]

motivo algumas plataformas propõem uma pré-seleção a fim de determinar o grau de conhecimento de cada participante e só então liberar ou não a tarefa para o participante [27].

Cabe lembrar que como decorrência da decomposição das tarefas é necessário realizar a adequação da documentação atribuída a cada uma das micro tarefas geradas, isto é, ao realizar a fragmentação da tarefa cada uma das micro tarefas deve possuir informação suficiente para permitir que as mesmas sejam realizadas. No entanto, a documentação não deve ser específica demais a ponto de limitar a inovação advinda da multidão (o que iria contra a um dos princípios do *crowdsourcing* que preconiza a inovação para as soluções), ou ampla demais ao ponto de o contratante não receber o resultado esperado. Logo, a atividade de decomposição das tarefas é um desafio considerável no que diz respeito à eficiência na resolução das mesmas (Figura 2.7), uma vez que através da decomposição e da qualidade da documentação das tarefas é possível melhorar o desempenho da multidão com o paralelismo na execução das tarefas [37].

No cenário contemporâneo, em que uma tarefa em uma plataforma de *software crowdsourcing* pode assumir diferentes formatos, seja um problema para ser resolvido, um modelo de inovação a ser proposto, uma questão para coleta de dados ou então uma iniciativa para angariar fundos, dentre outros, entende-se que o participante deverá possuir determinadas características para conseguir executá-la. Por exemplo: para uma tarefa que envolve a resolução de um problema o participante provavelmente necessitará de conhecimento ou experiência prévia, já em uma iniciativa para angariar fundos o participante precisará ter dinheiro. Conforme o tipo e/ou formato da tarefa será indispensável que o participante possua uma destas características: experiência, ideias, conhecimento, habilidades, tecnologias ou dinheiro [14].

De acordo com Hosseini et al. [14], a seguir são destacadas oito características relacionadas às tarefas neste modelo de desenvolvimento de software:

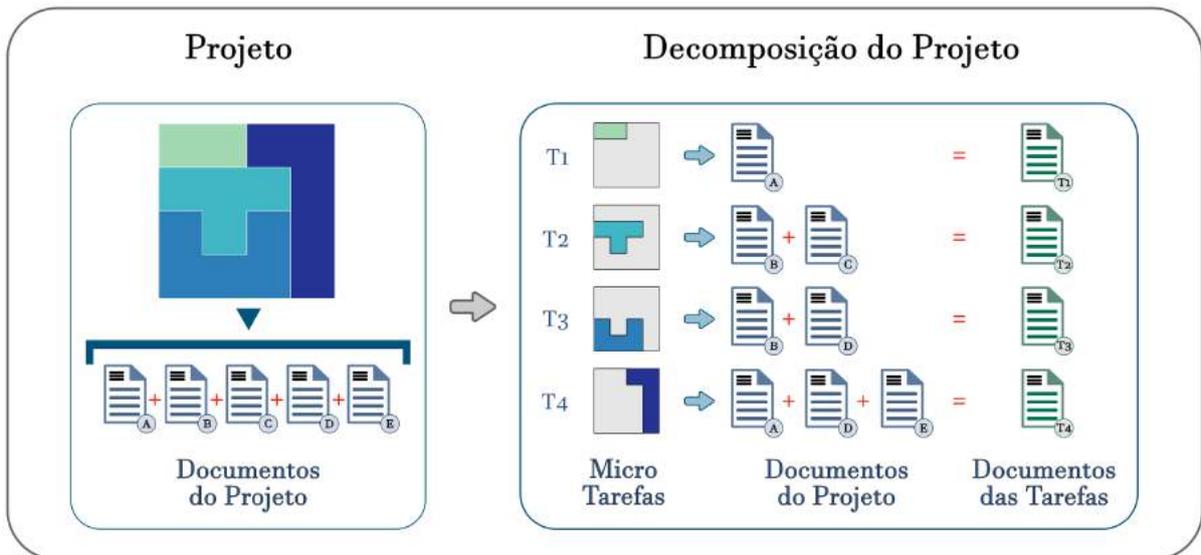


Figura 2.7 – Processo de decomposição das tarefas e documentação

Fonte: Elaborado pelo autor

- Operação tradicional:

Representa uma tarefa que pode ser realizada pela organização se não fosse *crowdsourced*, isto é, sem o conceito de *crowdsourcing* esta atividade seria realizada pelos colaboradores da organização ou seria terceirizada de forma tradicional.

- Tarefa de terceirização:

Representa uma tarefa que por motivos de redução de custos operacionais deverá ser realizada por mão-de-obra externa à organização.

- Modularidade:

Representa uma padronização de uma unidade de tamanho, *design*, construção, dentre outras que uma tarefa poderá ter. Uma tarefa de *crowdsourcing* preferencialmente deve ser atômica, mas geralmente é uma tarefa mais complexa que é decomposta em micro tarefas para serem realizadas pela multidão.

- Complexidade:

Identifica a condição ou a qualidade de uma tarefa ser complexa, isto é, possuir duas ou mais partes inter-relacionadas. Uma tarefa pode ser simples e atômica, ou pode ser complexa e possuir outras tarefas relacionadas a ela. A diferença entre Modularidade e Complexidade é que algumas tarefas podem ser complexas e atômicas ao mesmo tempo (não ter sido decompostas em micro tarefas). Já em outros casos uma tarefa simples pode ser decomposta em outras micro tarefas.

- Solucionável:

Representa a viabilidade que uma tarefa tem de ser resolvida. Uma tarefa de *crowdsourcing* geralmente é uma tarefa simples o suficiente para que um humano possa resolvê-la, mas complexa o bastante para que um computador não possa solucioná-la. Por exemplo, uma tarefa pode ser complexa e atômica, no entanto é simples o bastante para que um membro da multidão consiga completá-la.

- Características de automação:

A automação identifica que uma atividade possui um fluxo de ações padronizável e assim pode ser realizada de forma autônoma. Geralmente uma tarefa que é difícil de automatizar, seja por complexidade de execução ou pelo custo desta automatização, é uma tarefa ideal para o uso de *crowdsourcing*. Como mencionado, se uma tarefa é muito complexa para que os computadores possam executá-la, pressupõe-se que ela é difícil de automatizar.

- Orientado ao usuário:

Representam atividades que são controladas pelos usuários. Essas atividades podem ser apresentadas de três formas: (i) tarefas para resolução de problemas, o que significa que a multidão deve fornecer uma solução para um problema particular, (ii) tarefas de inovação, que significa que a multidão deve gerar ideias ou criar projetos, (iii) tarefas de co-criação, que significa que a multidão participa de um processo para desenvolver um produto.

- Tipo de Contribuição:

Uma tarefa de *crowdsourcing* pode ser apresentada de duas formas em relação ao tipo de contribuição gerado pela multidão: (i) pode ser uma contribuição individual, ou seja, cada participante da multidão executa a tarefa *crowdsourced* sem ajudar ou colaborar com outros elementos da multidão, (ii) pode ser uma contribuição colaborativa, ou seja, diferentes participantes da multidão realizam a tarefa atuando em equipe, compartilhando recursos e trabalhando em parceria.

2.4 Engenharia de Requisitos e Qualidade da Documentação

A Engenharia de Requisitos é um subgrupo de conhecimento da Engenharia de Software que está diretamente relacionada à definição de quais são as necessidades que o software deverá atender [40]. De acordo com Brooks [5], os requisitos são um fator crucial para o sucesso no desenvolvimento de um software.

“A parte mais difícil da construção de software é decidir precisamente o que construir. Nenhuma outra etapa do trabalho conceitual é tão difícil quanto estabelecer os requisitos técnicos detalhados, incluindo todas as interfaces com as pessoas (usuários finais), com as máquinas e com outros sistemas de software. Nenhuma outra parte do trabalho compromete tanto o software resultante se for executada

de forma errada. Nenhuma outra parte é mais difícil de ser retificada mais tarde.”
Traduzido de [5] (p.17)

Conforme esclarece Sommerville [35], os requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços que ele oferece e as restrições que ele deve possuir. Estes requisitos refletem as necessidades dos clientes para um sistema que serve a uma determinada finalidade. O processo de descobrir, analisar, documentar e verificar esses serviços e suas restrições é denominado de Engenharia de Requisitos. Em relação aos requisitos, estes podem ser classificados em três níveis básico de abstração: requisitos de negócio (*business requirements*), requisitos de usuários (*user requirements*) e requisitos funcionais (*functional requirements*). A partir desta classificação inicial é possível realizar uma segmentação mais detalhada.

No entendimento de Wiegers [40], a maneira como ocorre a classificação dos requisitos pode variar conforme a sua natureza ou objetivo, sendo os mais frequentes os requisitos de usuários, requisitos de software, requisitos de negócio, requisitos funcionais, requisitos de sistema, requisitos de produto, requisitos de projeto, história de usuários ou restrições. Cada um desses tipos de requisitos, bem como a sua definição são apresentados na Tabela 2.3 [40].

Pode-se destacar ainda outra abordagem que propõe a classificação dos requisitos sob a ótica da Análise de Negócios, apresentada pelo Guia BABOK 3 [18]. Esta classificação é descrita a seguir:

- Requisitos de negócios: declarações de objetivos e resultados que descrevem por que uma mudança foi iniciada. Eles podem abranger toda uma empresa, uma área de negócios ou uma iniciativa específica.
- Requisitos das partes interessadas: descrevem as necessidades das partes interessadas que devem ser atendidas para atingir os requisitos do negócio. Eles podem servir como uma ponte entre o negócio e os requisitos de solução.
- Requisitos de solução: descrevem os recursos e as qualidades de uma solução que atende aos requisitos das partes interessadas. Eles fornecem o nível adequado de detalhes para permitir o desenvolvimento e a implementação da solução. Os requisitos de solução podem ser divididos em duas subcategorias: (i) os requisitos funcionais, que descrevem os recursos que uma solução deve ter em termos de comportamento e informações que a solução gerenciará e (ii) os requisitos não funcionais ou requisitos de qualidade de serviço, que não se relacionam diretamente com o comportamento da funcionalidade da solução, mas descrevem condições em que uma solução deve permanecer efetiva ou qualidades que uma solução deve ter.
- Requisitos de transição: descreve os recursos que a solução deve ter e as condições que a solução deve atender para facilitar a transição do estado atual para o estado futuro, mas que não são necessárias após a conclusão da alteração. Eles são diferenciados de outros tipos de requisitos porque são de natureza temporária. Os requisitos de transição abordam tópicos como conversão de dados, treinamento e continuidade do negócio.

Tabela 2.3 – Tipos de requisitos

Tipo de Requisito	Definição
Requisito de Negócio (<i>Business requirement</i>)	Representa um objetivo de negócio de alto nível para a organização que produz ou define um sistema.
Regra de Negócio (<i>Business rule</i>)	Representa uma política, diretriz, norma ou regulamento que define ou restringe alguns aspectos do negócio. Não é um requisito de software em si, mas define melhor o seu escopo e pode dar origem de vários tipos de requisitos de software.
Restrição (<i>Constraint</i>)	Representa algum tipo de restrição que é imposta para o desenvolvimento de um produto.
Requisito de Interface (<i>External interface requirement</i>)	Descreve uma forma de comunicação entre um sistema e um usuário, outro sistema ou um dispositivo de hardware.
Funcionalidade (<i>Feature</i>)	Representa recursos do sistema logicamente relacionados que fornecem valor ao usuário e são descritos por um conjunto de requisitos funcionais.
Requisito Funcional (<i>Functional requirement</i>)	São as descrições de um comportamento que um sistema deve exibir conforme condições específicas pré-definidas.
Requisito Não Funcional (<i>Nonfunctional requirement</i>)	São as descrições de uma propriedade ou característica que um sistema deve exibir ou uma restrição que ele deve respeitar.
Atributo de Qualidade (<i>Quality attribute</i>)	Representa um tipo de requisito não funcional que descreve uma característica de desempenho de um produto.
Requisito de Sistema (<i>System requirement</i>)	Representa requisitos de alto nível para o produto que, em geral, estão ligados a dependência/acoplamento a outros itens que podem ser softwares ou hardwares.
Requisito de Usuário (<i>User requirement</i>)	Representa um objetivo ou tarefa que determinados usuários devem realizar com um sistema ou um atributo desejado para o produto.

Fonte: Adaptado de [40] (cap 1, p. 7)

Os tipos de requisitos apresentados possuem um forte relacionamento entre si, conforme exposto na Figura 2.8. Esta figura evidencia como os tipos de requisitos estão relacionados entre si e o contexto de sua utilização, considerando um *framework* padrão para a engenharia de requisitos representado através dos documentos produzidos e atribuídos a cada tipo de requisito.

Assim, com o objetivo de promover uma melhoria na qualidade dos requisitos obtidos para a construção de software, foi proposto um *framework* para apoiar a Engenharia de Requisitos. Este processo é descrito de forma simplificada a partir de quatro etapas: Elicitação, Análise, Especificação e Validação, conforme demonstrado na Figura 2.9. Há um forte relacionamento entre cada uma das etapas com o propósito de que existam pontos de controle/verificação e retorno à etapa anterior, o que demonstra que o *framework* propõe uma operação cíclica [40].

Cada uma destas etapas possui um conjunto de ações previstas a fim de se obter o melhor resultado na identificação e documentação dos requisitos. Estas etapas são definidas como segue:

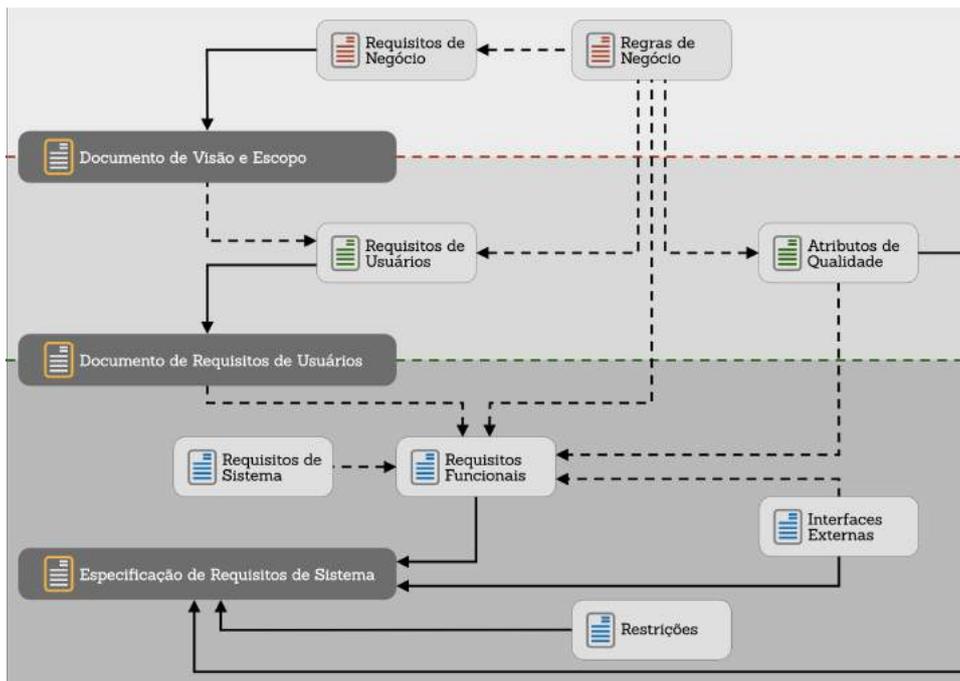


Figura 2.8 – Relacionamento entre tipos de requisitos

Fonte: Adaptado de [40] (cap. 1, p. 8)

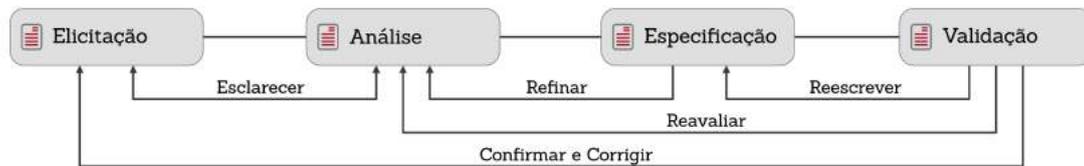


Figura 2.9 – Etapas do processo de desenvolvimento de requisitos

Fonte: Adaptado de [40] (cap. 3, p. 45)

- Elicitação: compreende as atividades para o entendimento inicial dos requisitos.
- Análise: envolve o detalhamento dos requisitos.
- Especificação: envolve a documentação e entendimento relacionado aos requisitos.
- Validação: consiste em verificar se a documentação satisfaz os objetivos de negócio.

Associado ao uso deste *framework* para a engenharia de requisitos existe um conjunto de boas práticas para cada uma das etapas do processo, conforme apresentado na Tabela 2.4 [40].

As práticas citadas na Tabela 2.4 são amplas na medida em que abrangem todo o processo da Engenharia de Requisitos. Uma vez finalizado este processo tem-se como resultado uma documentação de requisitos de acordo com as necessidades dos usuários-chave. A elaboração desta documentação deve ser muito cuidadosa e criteriosa a fim de registrar de forma correta e clara todas as informações necessárias para o desenvolvimento do requisito. Neste sentido, as características relacionadas à documentação dos requisitos envolvem a forma como devem ser registradas as informações, nível de detalhamento, entre outros. O que significa que a documentação tem sua origem

Tabela 2.4 – Boas práticas para a engenharia de requisitos

Elicitação
<ul style="list-style-type: none"> • Definir visão e escopo • Identificar classes de usuário • Selecionar produtos vencedores • Realizar grupos focais • Identificar as necessidades dos utilizadores • Identificar eventos e respostas do sistema • Realizar entrevistas de elicitación • Realizar oficinas de elicitación • Observar os usuários na execução de seus trabalhos • Distribuir questionários • Realizar análise de documentos • Examinar os relatórios de problemas • Reutilizar requisitos existentes
Análise
<ul style="list-style-type: none"> • Modelar o ambiente de aplicação • Criar protótipos dos requisitos • Analisar a viabilidade do requisito • Priorizar os requisitos • Criar um dicionário de dados • Modelar os requisitos • Analisar as interfaces • Atribuir requisitos aos subsistemas
Especificação
<ul style="list-style-type: none"> • Adotar modelos de documentos • Identificar origens de requisitos • Excepcionalmente etiquetar cada exigência • Registrar regras de negócios • Especificar requisitos não funcionais
Validação
<ul style="list-style-type: none"> • Revisar os requisitos • Testar os requisitos • Definir critérios de aceitação • Simular os requisitos

Fonte: Adaptado de [40] (cap. 3, p. 44)

no requisito e que deve expressar o objetivo a ser alcançado pelo mesmo. Os critérios de excelência para a documentação dos requisitos são apresentados na Tabela 2.5 [40]. Quando se faz referência a um conjunto de requisitos é necessário considerar alguns fatores específicos conforme apresentado na Tabela 2.6 [40].

De forma análoga a Wiegers [40], o Guia BABOK também destaca um conjunto de características inerentes aos requisitos e que são referenciadas em suas duas últimas versões: BABOK 2 e BABOK 3. Estas características estão descritas na Tabela 2.7, referente ao BABOK 2 [17] e na Tabela 2.8, referente ao BABOK 3 [18]. Convém esclarecer que o uso de duas versões do Guia BABOK ocorre em função da versão 3 agregar a Extensão Ágil, proposta pelo *International Institute of Business Analysis (IIBA)*.

Considerando que a principal função do requisito é transmitir as informações aos envolvidos no processo de construção de um software, independente se for um ator de negócio ou técnico, todos

Tabela 2.5 – Critérios individuais de excelência para documentar requisitos

Critério	Descrição
Completo (<i>Complete</i>)	Cada requisito deve conter todas as informações necessárias para que o leitor possa compreendê-lo. No caso de requisitos funcionais, isso significa fornecer as informações necessárias para que o desenvolvedor possa implementar corretamente o requisito. Quando não for possível definir um item é preciso deixá-lo explícito na documentação para evitar impedimento ou atraso por parte do desenvolvedor.
Preciso (<i>Correct</i>)	Cada requisito deve descrever com precisão sua funcionalidade a fim de viabilizar a sua construção. Com o objetivo de garantir a exatidão dos requisitos de usuário, estes devem revisá-los.
Exequível (<i>Feasible</i>)	Deve ser possível implementar cada requisito dentro das capacidades e limitações conhecidas do sistema e ambiente operacional, assim como dentro das limitações do projeto.
Necessário (<i>Necessary</i>)	Cada requisito deve descrever uma funcionalidade que agregue valor de negócio.
Priorizado (<i>Prioritized</i>)	Priorizar os requisitos do negócio de acordo com os que são mais importantes para alcançar o resultado desejado.
Sem ambiguidade (<i>Unambiguous</i>)	O requisito não deve ser ambíguo. A linguagem natural é propensa a dois tipos de ambiguidades. Uma quando é possível interpretar um requisito de mais de uma forma (indeterminação) ou quando um requisito é interpretado de forma diferente por pessoas diferentes (clareza).
Verificável (<i>Verifiable</i>)	Se um requisito não é verificável, decidir se ele foi corretamente implementado ou não se torna uma questão de opinião, não de análise objetiva. Requisitos que apresentam problemas de inconsistência, ambiguidade ou estão incompletos também não podem ser verificados.

Fonte: Adaptado de [40] (cap. 11, p. 204-205)

Tabela 2.6 – Critérios de excelência para documentar requisitos em conjunto

Critério	Descrição
Completo (<i>Complete</i>)	Nenhum requisito ou informação necessária deve estar ausente. Na prática, é muito difícil de documentar todos os requisitos para um sistema. Normalmente existem alguns requisitos assumidos ou implícitos que, com isso, trazem maior risco do que os requisitos explicitamente declarados.
Consistente (<i>Consistent</i>)	Requisitos consistentes não conflitam com outros requisitos do mesmo tipo ou com requisitos de negócios, usuários ou de sistema, independente do seu nível. Se os conflitos não forem resolvidos no momento da documentação dos requisitos este problema surgirá na etapa de desenvolvimento.
Modificável (<i>Modifiable</i>)	Os requisitos podem ser alterados ou atualizados, mas sempre mantendo um histórico das alterações realizadas. A modificação em um requisito está diretamente relacionada a rastreabilidade do mesmo, uma vez que os ajustes devem prever a correção de todos os requisitos impactados.
Rastreável (<i>Traceable</i>)	Um requisito que possui rastreabilidade demonstra a possibilidade de identificar sua relação de ascendência, descendência e/ou correlação com outros requisitos ou itens de documentação produzidos.

Fonte: Adaptado de [40] (cap. 11, p. 205)

Tabela 2.7 – Características de qualidade de um Requisito - Visão BABOK 2

Característica	Descrição
Coesão	Um conjunto coeso de requisitos está relacionado a apenas uma coisa, seja ela um processo de negócio, regra de negócio, unidade organizacional, etc. Todos os requisitos, em um conjunto ou modelo, devem apoiar o seu propósito e escopo geral
Compleitude	O conjunto total de requisitos deve representar todos os requisitos relevantes. Além disso, todo requisito individual deve ser completo. Garantia de que cada requisito seja autocontido, sem nenhuma informação faltante
Consistência	Garantia de que os requisitos individuais não conflitem entre si ou que descrevam o mesmo requisito utilizando palavreado diferente. Além disso, o nível de detalhes fornecido por cada requisito, em um conjunto ou modelo, deve ser o mesmo
Correção	Defeitos nos requisitos levarão a defeitos na solução resultante
Viabilidade	Cada requisito deve ser implementável dentro da infraestrutura existente, dentro do orçamento existente, prazo e recursos disponíveis para a equipe (ou o projeto deve desenvolver a capacidade para implementar o requisito). O analista de negócios precisa trabalhar junto à equipe do projeto para tomar essas decisões
Ajustabilidade/adaptação	Requisitos inter-relacionados devem ser mantidos agrupados a fim de se tornem modificáveis. Essa característica é demonstrada através de uma estruturação lógica dos requisitos
Não Ambiguidade	Requisitos individuais nunca podem ser pouco claros. Um requisito não pode permitir a formação de múltiplas e divergentes interpretações válidas
Testabilidade	Deve haver uma forma de provar que um requisito foi atendido. Cada requisito deve ser testável – isto é, deve ser possível desenhar um teste que possa ser usado para determinar se uma solução atendeu ao requisito, ou utilizar algum outro meio de determinar o aceite, ou não, de uma solução que atende ao requisito

Fonte: Adaptado de [17]

devem estar alinhados com o objetivo do requisito. Com frequência trata-se do requisito como um “documento escrito”, entretanto a documentação pode ser mais ampla, envolvendo outras formas de registro do conhecimento, como diagramas, protótipos, vídeos, entre outros. Este processo de documentar os requisitos deve ter como premissa dois pontos fundamentais [40]:

- Todos que leem o requisito chegam à mesma interpretação, independente do leitor.
- A interpretação de cada leitor coincide com o que o autor pretendia comunicar.

Cabe destacar que segundo o Guia BABOK 2 [17] um requisito textual deve possuir um conjunto de características que o definam e permitam a sua interpretação, tais como: a capacidade

Tabela 2.8 – Características de qualidade de um Requisito - Visão BABOK 3

Característica	Descrição
Atômico	O requisito deve ser entendido independentemente de outros requisitos ou projetos, ser autocontido em sua descrição.
Completo	O requisito deve possuir um nível de detalhe adequado em sua descrição, ou seja, seja suficiente para orientar o trabalho a ser realizado. O nível de completude necessário varia conforme a perspectiva ou metodologia utilizada, bem como o ponto do ciclo de vida em que o requisito se encontra.
Consistente	O requisito deve estar alinhado com as necessidades identificadas pelas partes interessadas e não apresentar conflito com outros requisitos.
Conciso	Não deve conter conteúdo desnecessário na documentação do requisito.
Viabilidade	O requisito deve ser factível considerando os riscos, cronograma e orçamento acordados, ou ser considerado viável o suficiente para uma investigação através de experimentos ou protótipos,
Sem ambiguidade	O requisito deve estar claramente descrito de forma a deixar claro o que deve ser feito e assim permitir avaliar se a solução proposta satisfaz ou não o requisito.
Testável	O requisito deve permitir a sua verificação com o objetivo de avaliar se o requisito ou design foi cumprido. Os níveis aceitáveis de verificação dependem do nível de abstração do requisito ou design.
Priorizado	O requisito deve estar devidamente classificado e negociado em termos de importância e valor em relação a todos os outros requisitos.
Compreensível	O requisito deve ser descrito conforme a terminologia utilizada pelos entes envolvidos no projeto.

Fonte: Adaptado de [18]

de apresentar uma solução, as condições que devam existir para o requisito operar e as restrições que possam impedir que a solução atenda ao requisito. Nesse sentido, são listadas abaixo algumas orientações propostas para que se escreva um requisito textual:

- Expressar um e somente um requisito de cada vez.
- Evitar cláusulas condicionais complexas.
- Não assumir que o seu leitor possui conhecimento do tema tratado.
- Usar terminologia consistente.
- Expressar requisitos como um verbo ou frase verbal.
- Escrevê-los em voz ativa, descrevendo claramente quem ou o quê é responsável por atender ao requisito.

- Usar terminologia familiar às partes interessadas que devem revisar ou usar o requisito.

Em outras palavras, ao final do processo de mapeamento e documentação de um requisito, se o mesmo estiver adequado às boas práticas, critérios de excelência e premissas anteriormente apresentados, a tendência é que o requisito seja implementado com sucesso.

3. METODOLOGIA DE PESQUISA

O presente capítulo apresenta a metodologia de pesquisa aplicada a este estudo. Na Seção 3.1, é apresentado o método de pesquisa definido para este trabalho, na Seção 3.2, é exposta a visão geral desta metodologia através do desenho da pesquisa. Na Seção 3.3, é descrita a Revisão da Literatura. Já nas Seções 3.4 e 3.5, são apresentados os dois métodos aplicados para esta pesquisa qualitativa que são o Estudo de Caso e Estudo de Campo, respectivamente. Na Seção 3.6, está descrita a técnica de análise de conteúdo utilizada nesta pesquisa.

3.1 Método de Pesquisa

A metodologia de pesquisa visa descrever como o trabalho foi realizado indicando seu método, critérios e protocolos a fim de se obter um resultado significativo e relevante ao tema investigado. Conforme Creswell [6], a escolha do método que será utilizado para uma pesquisa é crucial para a definição das demais etapas. Apesar dos métodos quantitativo e qualitativo apresentarem processos similares, a definição prévia de qual modelo será utilizado permite a adequação da abordagem e de instrumentos mais apropriados a cada um destes métodos. Neste sentido, optou-se por utilizar o Método Qualitativo para o desenvolvimento desta pesquisa, pois acredita-se que este método é mais adequado aos objetivos deste estudo.

Para compreender o motivo da seleção do método qualitativo é importante destacar alguns de seus conceitos e características. A seguir é apresentado um conceito sobre pesquisa qualitativa proposto por Creswell [7].

“A pesquisa qualitativa começa com pressupostos e o uso de estruturas interpretativas/teóricas que informam o estudo dos problemas da pesquisa, abordando os significados que os indivíduos ou grupos atribuem a um problema social ou humano. Para estudar esse problema, os pesquisadores qualitativos usam uma abordagem qualitativa da investigação, a coleta de dados em um contexto natural sensível às pessoas e aos lugares em estudo e a análise dos dados que é tanto intuitiva quanto dedutiva e estabelece padrões ou temas. O relatório final ou a apresentação incluem as vozes dos participantes, a reflexão do pesquisador, uma descrição complexa e interpretação do problema e a sua contribuição para a literatura ou um chamado à mudança.”

Adaptado de [7]

Dentre as características do método qualitativo propostos por Creswell [6, 7] ressaltam-se:

- Ambiente natural: a pesquisa acontece onde os participantes vivenciam o problema ou questão de pesquisa, isto é, o pesquisador vai a campo coletar os dados.
- O pesquisador como um instrumento fundamental: o pesquisador faz a coleta pessoalmente por meio de análise de documentos, questionários, entrevistas ou observação do comportamento dos participantes. Não é comum utilizar instrumentos construídos por outros pesquisadores.
- Múltiplas fontes de dados: geralmente a pesquisa ocorre com a utilização de mais de uma fonte de coleta de dados com o objetivo de permitir a triangulação de dados.

- Análise de dados indutiva: os pesquisadores criam seus próprios padrões, categorias ou temas para organizar os dados conforme o tema e a evolução da pesquisa.
- Significado para os participante: o foco do pesquisador é a aprendizagem do significado que os participantes dão ao problema ou questão de pesquisa e não dos conceitos apresentados na literatura.
- Projeto emergente: significa que o projeto não pode ser rigidamente definido, ou seja, em todas as etapas do processo mudanças podem ser necessárias uma vez que o pesquisador em campo pode identificar ajustes indispensáveis para o desenvolvimento da pesquisa.
- Lente teórica: com frequência o pesquisador utiliza “lentes” para ver o objeto de estudo a fim de adequar ao contexto de pesquisa.
- Interpretativo: por natureza este tipo de investigação visa apresentar a interpretação do pesquisador sobre os fatos coletados ou observados dentro do seu contexto.
- Relato holístico: o pesquisador procura apresentar um quadro do problema ou questão de pesquisa com a sua interpretação sobre a perspectiva dos participantes.

As características do método qualitativo propostas por Yin [43] são listadas a seguir:

- Estudar o significado da vida das pessoas nas condições da vida real;
- Representar as opiniões e perspectivas das pessoas (participantes) de um estudo;
- Abranger as condições contextuais em que as pessoas vivem;
- Contribuir com revelações sobre conceitos existentes ou emergentes que podem ajudar a explicar o comportamento social humano; e
- Esforçar-se por usar múltiplas fontes de evidência em vez de se basear em uma única fonte.

Assim, a escolha pelo método qualitativo teve como propósito o alinhamento aos objetivo de pesquisa que é investigar como a documentação das tarefas disponibilizadas na plataforma TopCoder é utilizada na prática e quais as percepções que os participantes possuem ao utilizar esta documentação. Nesse sentido, pode-se perceber que várias das características do método qualitativo são intrínsecas ao tema de pesquisa. Salienta-se a necessidade de realizar uma avaliação em condições reais para obter as percepções dos participantes das tarefas em relação à documentação proposta pela plataforma TopCoder. Em outras palavras, para o desenvolvimento desta investigação foi necessário ir a campo para verificar como os participantes percebem e avaliam a documentação das tarefas de modo a possibilitar melhorias em sua estrutura. Para tanto, foram escolhidos como método para a pesquisa o Estudo e Caso e o Estudo de Campo.

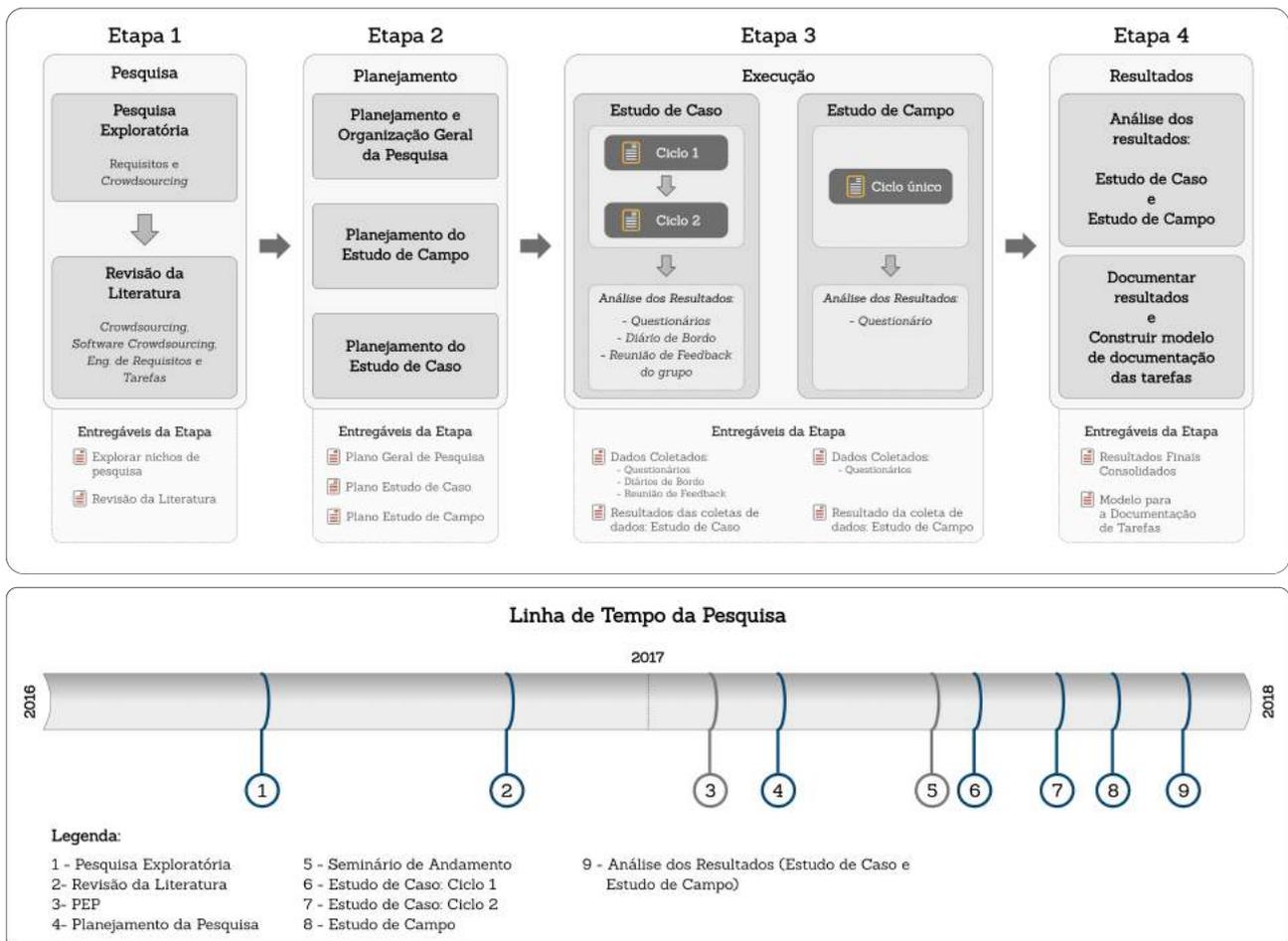


Figura 3.1 – Representação do Desenho da Pesquisa - Visão Geral

Fonte: Elaborado pelo autor

3.2 Desenho da Pesquisa

Nesta seção do trabalho é apresentado o desenho da pesquisa conforme representado na Figura 3.1. Esta representação descreve cada uma das etapas realizadas neste estudo.

- **Etapa 1: Pesquisa Exploratória e Revisão da Literatura**
Esta etapa do estudo teve como objetivo avaliar nichos de pesquisa relacionados à Engenharia de Requisitos. A partir desta premissa foi realizada uma pesquisa exploratória a fim de verificar eventuais lacunas sobre o tema. Com os resultados obtidos neste processo foi possível identificar um tema de pesquisa que relaciona a Engenharia de Requisitos e o *Software Crowdsourcing*. Com o tema definido foi realizada uma revisão da literatura cujo objetivo foi verificar como este tema está sendo abordado nas produções científicas atuais. Os detalhes referentes a revisão da literatura e os resultados completos podem ser encontrados no trabalho de Vaz [38].
- **Etapa 2: Planejamento da Pesquisa**
Com a definição do tema de pesquisa foi possível realizar o planejamento das demais etapas

necessárias para a realização deste estudo. Pode-se dividir esta etapa em três grandes momentos: (i) plano geral da pesquisa, cujo propósito foi determinar uma visão completa de todos os pontos que deveriam ser realizados e as diretrizes necessárias a sua realização, (ii) plano para o Estudo de Caso, que buscou determinar as etapas necessárias ao Estudo de Caso e a definir suas diretrizes. Os detalhes desta etapa são descritos na Seção 3.4 deste capítulo, e (iii) plano para o Estudo de Campo, que buscou determinar como a coleta de dados deveria ser realizada nesta investigação. Os detalhes desta etapa são descritos na Seção 3.5 deste capítulo.

- **Etapa 3: Execução da Pesquisa**

Foi nessa etapa que efetivamente ocorreram as ações de coleta de dados para esta pesquisa. Para o Estudo de Caso foram construídos os instrumentos de coleta de dados, que foram validados e aplicados aos participantes. Os detalhes deste processo podem ser verificados no Capítulo 5. Já os procedimentos realizados para o desenvolvimento do Estudo de Campo têm seu detalhamento apresentado no Capítulo 6 deste trabalho.

- **Etapa 4: Análise dos Resultados Parciais**

A análise dos resultados do Estudo de Caso e do Estudo de Campo foi efetuada com base na consolidação das respostas obtidas através dos participantes e pelo acompanhamento de suas atividades conforme o método utilizado. O detalhamento dos resultados de cada método pode ser observado na Seção 5.3 para o Estudo de Caso e na Seção 6.3 para o Estudo de Campo.

- **Etapa 5: Resultados e Recomendações**

Ao consolidar as informações obtidas a partir de cada um dos métodos utilizados na pesquisa foi possível realizar uma ampla análise e assim identificar pontos de convergência referentes às percepções/opiniões apresentadas pelos participantes dos estudos. Fundamentado nestes resultados foi proposto um modelo de documento para as tarefas disponibilizadas pela plataforma TopCoder. Os detalhes sobre os resultados finais e a proposta estão descritos no Capítulo 7.

3.3 Revisão da Literatura

Nesta seção é apresentada a metodologia utilizada para a Revisão da Literatura. Esta técnica visa estabelecer um conjunto de critérios claros e específicos aplicados ao processo de pesquisa com o objetivo de identificar como um tema de estudo está sendo contemplado nas produções científicas de uma determinada área. Através do mapeamento e análise do material científico produzido sobre um tema específico é possível identificar tendências e/ou lacunas em uma área de pesquisa. Com isso, viabiliza-se o suporte teórico para novas pesquisas, além de tornar possível a sua reprodução para validação e para o avanço dos estudos na área investigada. A revisão de literatura é uma prática recorrente em diversas áreas, como saúde, educação, dentre outras, e seu uso vem crescendo

na área da Engenharia de Software. A representação da forma adaptada do processo de revisão de literatura da área da saúde para a área da Engenharia de Software foi elaborada por Kitchenham [21] e Biolchini [3].

Segundo estudos desenvolvidos por Biolchini [3], os principais benefícios que este método traz para a pesquisa podem ser resumidos da seguinte forma:

- Identificação das evidências objetivas e relevantes para o tema de pesquisa;
- Reconhecimento através da análise do material pesquisado de quais são as lacunas existentes para melhor direcionar trabalhos futuros; e
- Obtenção de uma visão ampliada sobre o tema, registrando no protocolo de pesquisa de forma que se viabilize a reprodução da pesquisa;

De acordo Kitchenham [21], o método de revisão da literatura deve utilizar o material científico relevante disponível relacionado ao tema investigado. O processo deve ser realizado de forma rigorosa utilizando diferentes materiais, como livros, artigos, anais de congressos, periódicos, dentre outros materiais que sejam aplicados à área de pesquisa. Ao consolidar estes conhecimentos é possível traçar as relações entre os diversos artefatos e consolidar as informações.

Para o desenvolvimento de uma revisão da literatura é necessário definir o protocolo de pesquisa com as etapas e requisitos necessários ao processo. A construção do protocolo de pesquisa utilizado neste estudo teve como base os trabalhos de Biolchini [3] e Kitchenham [21]. Biolchini representa o processo em um macro cenário para aplicação da revisão da literatura, conforme pode ser observado na Figura 3.2.

Em contrapartida, Kitchenham [21] detalha as etapas do processo para a revisão da literatura em três fases, são elas: (i) Fase de Planejamento, nesta fase é proposto identificar os objetivos da pesquisa e definir o protocolo de pesquisa, (ii) Fase de Execução, é a fase em que ocorre efetivamente a revisão da literatura, ou seja, são realizadas as consultas as bases de artigos, refinamentos necessários à identificação dos artigos até a identificação de um conjunto de artigos relevantes ao tema de pesquisa proposto, e (iii) Fase de Resultados, esta é última etapa da revisão da literatura, onde ocorre a verificação de como os artigos estão alinhados ao tema de pesquisa, são identificadas as eventuais lacunas de pesquisa e são relatados os resultados obtidos com a revisão da literatura.

Os detalhes referentes a revisão da literatura e os resultados completos podem ser encontrados no trabalho de Vaz [38].

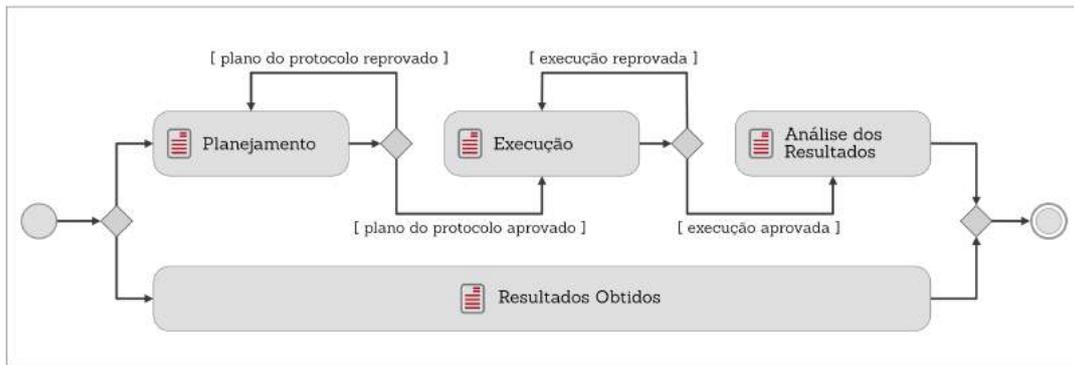


Figura 3.2 – Processo de Revisão da Literatura

Fonte: Adaptado de [3].

3.4 Estudo de Caso

De acordo com Creswell [7], na abordagem qualitativa é proposto que o pesquisador se situe no mundo real e assim possa coletar fatos/dados reais, obtidos através de diversos instrumentos como a observação, notas de campo, entrevistas e questionários. Essa abordagem permite ao pesquisador compreender e interpretar os fenômenos em termos de significados para as pessoas que estão envolvidas. Tendo em vista que esta pesquisa se caracteriza por ser qualitativa, uma das formas de “materializar” o problema de pesquisa no mundo real pode ser efetivada por meio de um Estudo de Caso.

No entendimento de Yin [43], a definição para Estudo de Caso pode ser segmentada em duas partes: a primeira parte trata do escopo do Estudo de Caso e a segunda parte visa caracterizar o Estudo de Caso propriamente dito, conforme exposto a seguir:

1. O Estudo de Caso é uma investigação empírica que:

- investiga um fenômeno contemporâneo (o “caso”) em profundidade e em seu contexto de mundo real;
- os limites entre o fenômeno e o contexto podem não ser claramente evidentes.

2. A investigação do Estudo de Caso:

- enfrenta uma situação tecnicamente diferenciada em que existirão mais variáveis de interesse do que pontos de dados;
- o resultado conta com múltiplas fontes de evidências, com os dados precisando convergir de maneira triangular;
- beneficia-se de desenvolvimentos anteriores e proposições teóricas para orientar a coleta e a análise de dados.

Segundo Gil [10], apesar de diversas definições existentes para o Estudo de Caso é possível destacar um conjunto de características que identificam este método de pesquisa, são elas:

- a É um delineamento de pesquisa. Não pode ser confundido com técnica, estratégia ou tática de coleta de dados;
- b Preserva o caráter unitário do fenômeno pesquisado. O evento é interpretado de forma unitária, independente de o caso em questão envolver um ator (indivíduo) ou grupo de atores (uma comunidade, uma organização, dentre outros);
- c Investiga um fenômeno contemporâneo. Mesmo podendo envolver um caráter histórico, o objeto de estudo ou fenômeno deve ocorrer no mesmo momento em que a pesquisa acontece;
- d Não separa o fenômeno do contexto. Não existe a restrição deliberada das variáveis estudadas por conta de viabilizar sua realização;
- e É um estudo em profundidade. A estrutura dos artefatos utilizados tendem a captar com maior profundidade os dados oferecidos pelos participantes; e
- f Requer a utilização de múltiplos procedimentos de coleta de dados. Pressupõe o uso de múltiplas formas para coletar os dados a fim de se obter um amplo conjunto de informações, permitindo assim realizar a triangulação destas informações.

Assim, optou-se por realizar um Estudo de Caso tendo em vista todas as características deste método já expostas anteriormente e especialmente pelos objetivos desta pesquisa, que visam investigar a documentação das tarefas disponibilizadas na plataforma TopCoder e os elementos que devem ser considerados na documentação de uma tarefa. Salienta-se em especial o seu propósito de avaliar as particularidades de um grupo específico dentro de um contexto de mundo real, no qual o pesquisador fez o acompanhamento dos participantes durante a análise e execução das tarefas. Desta forma, para o desenvolvimento desta investigação foram planejadas ações que são detalhadas no Capítulo 5 deste trabalho.

3.5 Estudo de Campo

Segundo McGrath [29], o ato de “fazer pesquisa” representa o uso sistemático de ferramentas teóricas e empíricas com o objetivo de compreender o fenômeno ou evento em estudo. Neste sentido, o Estudo de Campo tem como finalidade proporcionar ao pesquisador uma visão mais objetiva a respeito das ações e comportamentos das pessoas em seu habitat natural, ou seja, traz a pesquisa ao contexto real das ações das pessoas. Este tipo de pesquisa, que envolve contextos relacionados às ciências sociais e comportamentais, pode ser segmentado em três domínios distintos, embora inter-relacionados, indicados a seguir:

- Domínio *Substantivo*: neste domínio são identificados os elementos que são o foco do estudo e suas relações entre si e com o meio em que estão inseridos. Para as ciências sociais e comportamentais o interesse do estudo se dá em relação aos fenômenos e ações realizadas

pelos sistemas humanos (indivíduos, grupos, organizações, comunidades e similares). Isso significa que o estudo ocorre sobre os atores que estão vinculados ao contexto de pesquisa e a forma que atuam neste contexto.

- Domínio *Conceitual*: neste domínio os elementos fundamentais são as propriedades e ações que os sistemas humanos possuem e como eles se relacionam com os demais elementos do contexto, ou seja, refere-se às várias formas possíveis que os elementos de estudo se relacionam nos mais diversos graus e esferas, seja ele lógico, temporal ou casual. Estas relações é que importam neste domínio.
- Domínio *Metodológico*: neste domínio o fator fundamental é a metodologia aplicada para a obtenção das informações necessárias ao tema de estudo. Os métodos devem estar adequados ao foco de pesquisa a fim de se obter os dados desejados. O pesquisador deve adequar os instrumentos utilizados de forma a melhor garantir a coleta dos dados relacionados ao seu tema. Dentre os métodos disponíveis para medir e extrair estas informações pode-se utilizar um questionário, uma escala de classificação, um teste de personalidade, instrumentos de observação e gravação, técnicas para avaliar a qualidade de alguns produtos resultantes de desempenho de tarefas individuais ou grupais e similares.

No processo de coletar evidências é importante sempre verificar os seguintes aspectos: (i) quem, ou seja quais são os atores envolvidos, (ii) quais comportamentos e (iii) quando e onde, isto é, em quais contextos o evento em estudo esta envolvido, principalmente nas ciências sociais e comportamentais. Nesse sentido, conforme indicado por McGrath [29], quando da realização da coleta de dados é frequente a necessidade de maximizar as três características que envolvem este tipo de pesquisa, como apontado a seguir:

- A Generalização da evidência sobre as populações de atores foco da pesquisa.
- B Precisão da medição dos comportamentos que estão sendo estudados (precisão de controle sobre fatores estranhos que não estão sendo estudados).
- C Realismo da situação ou do contexto dentro do qual a evidência é coletada em relação aos contextos aos quais você deseja que suas evidências se apliquem.

Quando da realização de uma pesquisa, almeja-se maximizar as características anteriormente citadas. No entanto, isso é difícil de ser alcançado, visto que apesar destas características serem inter-relacionadas, ao maximizar uma delas outra será influenciada negativamente. Logo, o pesquisador deve equilibrar o nível de exigência em cada um destes aspectos e adequá-los a sua necessidade de pesquisa. Por exemplo: ao realizar uma pesquisa é possível tentar aumentar a precisão (item B) em relação aos dados obtidos (mensurar o comportamento e as variáveis relacionadas). Para isso o pesquisador poderá conduzir um experimento de laboratório cuidadosamente controlado. Ocorre que esta opção tenderá a interferir no resultados do evento estudado, podendo reduzir sua

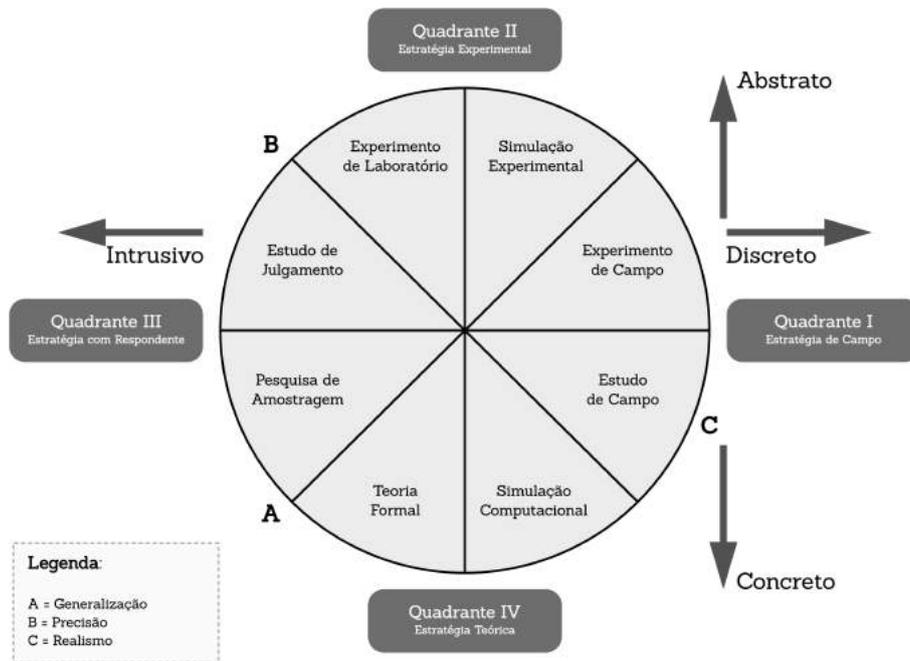


Figura 3.3 – Representação dos modelos de pesquisa de McGrath [29]

Fonte: Adaptado de [29]

“naturalidade” ou realismo das respostas, isto implica em reduzir o item C e talvez possa ainda implicar em redução na gama de atores envolvidos no estudo que incidirá no item A.

Este dilema de equacionar as estratégias de forma a obter os resultados mais adequados pode ser observado na Figura 3.3, que demonstra as principais estratégias de pesquisas utilizadas nas ciências sociais e comportamentais. Esta figura apresenta um conjunto de oito estratégias de pesquisa, ou configurações para a coleta de informações de pesquisa. Estas estratégias são apresentadas em um arranjo circular em relação a duas dimensões: (i) Abstrato versus Concreto, e (ii) Intrusivo versus Discreto. O posicionamento em relação a estratégia entre as duas dimensões varia conforme o grau de configuração utilizada na seleção e adequação das estratégias pelo pesquisador.

A construção proposta por McGrath [29] em seu diagrama demonstra como cada uma das estratégias de pesquisa são mais ou menos aderentes às características anteriormente citadas. A “gradação” relativa a cada uma destas estratégias varia conforme a sua natureza e objetivos, ou seja, uma estratégia ao ser definida como mais Abstrata e Discreta irá se afastar das características de Concreto e Intrusivo. Por exemplo, as quatro estratégias indicadas na parte superior do gráfico envolvem procedimento mais evasivos (Abstratos), ou seja, acabam por refletir nas ações dos sujeitos da pesquisa que estão em estudo. Em contrapartida, as quatro estratégias inferiores do gráfico possuem foco mais concreto em suas ações.

O gráfico proposto por McGrath apresenta como cada uma das três características podem ser maximizada em relação as estratégias propostas. O Critério A, a generalização relativa à população dos atores envolvidos na pesquisa pode ser maximizada através da Pesquisa de Amostragem ou Teoria Formal. Para o Critério B, relativo à precisão em relação à medição e o controle dos comportamentos pode ser maximizado com o uso de estratégias como Experiência de Laboratório e

Estudo de Julgamento. Por fim, o Critério C pode ser maximizado através da estratégia de Estudo de Campo. Desse modo, como não é possível maximizar todos os três critérios simultaneamente, percebe-se que as estratégias possuem limitações naturais de sua execução quando observadas pela perspectiva das suas características.

Outro aspecto do gráfico apresentado por McGrath faz referência aos quadrantes a que cada estratégia de pesquisa esta associada. Estas estratégias são agrupadas em pares e associadas aos quadrantes, como pode ser observado na Figura 3.3. A seguir são descritos estes quadrantes:

- **Quadrante I (Estratégia de Campo):** contém as estratégias de pesquisa que envolvem a observação do comportamento dos atores em condições naturais.
- **Quadrante II (Estratégia Experimental):** contém as estratégias de pesquisa que abordam as configurações de experimentos para a comprovação de uma ideia.
- **Quadrante III (Estratégia com Respondente):** contém as estratégias de pesquisa que envolvem a coleta de respostas dos participantes.
- **Quadrante IV (Estratégia Teórica):** contém estratégias de pesquisa que são teóricas, e não de caráter empírico.

Tendo em vista que esse estudo propõe-se a investigar a documentação das tarefas disponibilizadas na plataforma TopCoder e os elementos que devem ser considerados na documentação de uma tarefa, faz-se necessário utilizar métodos e instrumentos que melhor se ajustem a este objetivo. Portanto, a escolha pelo Estudo de Campo esta devidamente alinhada a esta necessidade, uma vez que a sua característica principal é a obtenção das informações mais reais possíveis dentro do contexto da pesquisa. Salienta-se que a seleção deste método é necessária à pesquisa na medida em que os participantes deste processo não vão realizar efetivamente as tarefas, mas sim avaliá-las a partir de suas experiências e conhecimentos. Com base nesta premissa, o planejamento para a realização do Estudo de Campo foi elaborado e será detalhado no Capítulo 6 deste trabalho.

3.6 Análise de Conteúdo

Considerando que este trabalho é de natureza qualitativa e os instrumentos utilizados para coletar as informações dos participantes foram os questionários, os diários de bordo e os documentos disponíveis na plataforma TopCoder, a utilização de uma técnica de análise de conteúdo é fundamental para consolidar todos os dados obtidos. Desta forma esta pesquisa utilizou a técnica de análise de conteúdo proposta por Bardin [2].

Como forma de compreender este processo é necessário perceber sua estrutura, assim segundo Gil [10], é possível descrever a análise de conteúdo como uma técnica analítica para estudar a comunicação humana de forma sistemática e objetiva. Pode-se destacar alguns dos objetivos desta técnica a seguir:

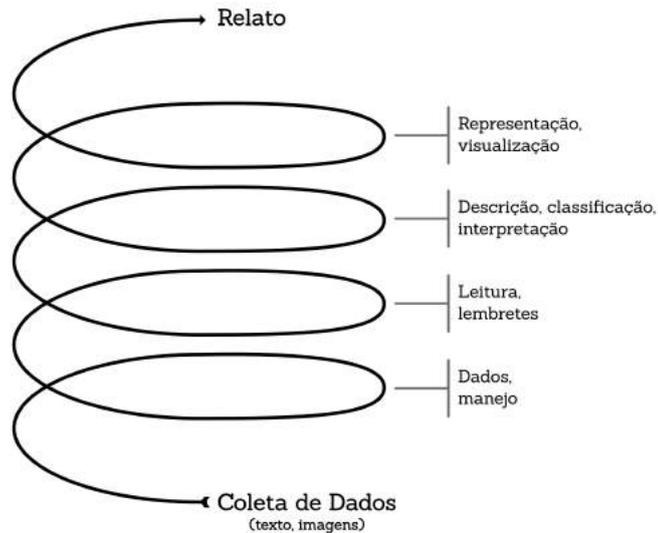


Figura 3.4 – Espiral evolutiva da análise de conteúdo de Creswell [7]

Fonte: Adaptado de [7]

- 1 Auxiliar na identificação de intenções e características dos comunicadores;
- 2 Identificar o *status* de pessoas ou grupos;
- 3 Revelar atitudes, interesses, crenças e valores de pessoas ou grupos;
- 4 Identificar o foco de atenção das pessoas ou grupos; e
- 5 Descrever as atitudes e respostas aos meios de comunicação.

Creswell [7] descreve sucintamente o processo de análise de conteúdo através da espiral de análise de dados, onde cada camada evolui sobre o conhecimento e estratificação da camada anterior, conforme demonstrado na Figura 3.4. Estas camadas são detalhadas a seguir:

- Dados, manejo: esta etapa corresponde ao momento inicial do processo de análise de conteúdo, no qual o pesquisador organiza e consolida seus dados de modo a ser mais eficiente na verificação e consulta dos dados, como também de modo mais adequado ao objetivo da pesquisa.
- Leitura, lembrete: esta etapa corresponde ao momento de imersão do pesquisador na análise dos dados coletados e consolidados de forma a construir um conhecimento mais consistente em relação às suas informações.
- Descrição, classificação, interpretação: nesta etapa o pesquisador, já de propriedade dos dados coletados e consolidados, poderá fazer o uso da codificação dos textos. Esta é considerada uma das etapas mais importantes do processo de análise de conteúdo.

- Representação, visualização: esta é a última etapa da espiral, neste momento o pesquisador já com um conhecimento estratificado em relação aos dados coletados e devidamente classificados poderá verificar qual a melhor forma de demonstrar os resultados obtidos, seja de forma textual, gráficos, tabelas, infográficos dentre outras representações.

De forma mais detalhada, Bardin [2] descreve a técnica de análise de conteúdo, que possui uma estrutura bastante semelhante a que foi apresentada por Creswell. A autora estabelece três etapas cronológicas para a análise de dados, conforme exposto a seguir:

- a A primeira etapa é a pré-análise, é o momento de organização dos materiais a serem analisados, da formulação das hipóteses e dos objetivos, e da elaboração de indicadores que fundamentem a interpretação final. Nessa etapa é realizada a leitura flutuante e a classificação do material coletado.
- b Na segunda etapa, a referida autora propõe a exploração do material em profundidade, e vincula o sucesso dessa fase ao trabalho desenvolvido na primeira etapa. É o momento de contagem, classificação e seleção das categorias.
- c A terceira e última etapa consiste no tratamento dos resultados, na inferência e na interpretação dos dados. Para Bardin [2], o pesquisador, de posse de dados tratados, “significativos e fiéis, pode então, propor inferências e adiantar interpretações a propósito dos objetivos previstos, ou que digam respeito a outras descobertas inesperadas”.

Considerando que os instrumentos utilizados para coletar os dados dos participantes foram questionários utilizados no Estudo de Caso e Estudo de Campo, Diários de Bordo utilizados no Estudo de Caso e documentos complementares, obtidos no decorrer da pesquisa (relatos apresentados pelos participantes e/ou documentação das tarefas disponíveis na plataforma TopCoder), todo este material foi examinado sob a perspectiva da análise de conteúdo proposta por Bardin [2]. Como resultado da etapa de pré-análise, os dados foram classificados levando-se em consideração os objetivos e o problema da pesquisa. O processo de análise foi aplicado à todos os instrumentos e documentos coletados e seguiu as premissas já citadas para o processo de análise de conteúdo.

Na etapa seguinte, com base no trabalho já desenvolvido e na leitura exaustiva do material, foi realizada a classificação e seleção das categorias. O processo de “construção” das categorias teve como propósito estabelecer relações entre os materiais coletados e o referencial teórico e deu-se de modo incremental, ou seja, foram realizados diversos ciclos em que as categorias foram revisitadas e ajustadas de forma a se obter um conjunto claro e relevante para a pesquisa. Inicialmente as classes emergiram de forma direta, como por exemplo a categoria “Clareza da documentação”, que nos primeiros ciclos estava associada de forma macro a itens que envolviam melhorias no entendimento da documentação da tarefa, mas que após alguns ciclos foram derivadas em outras categorias como “Detalhamento da Tarefa” e “Documentação Complementar”. Ao final de todo o processo, foi possível definir as categorias de análise que expressassem, da modo mais apropriado, as ideias provenientes do material analisado.

4. REVISÃO DA LITERATURA

Neste capítulo é apresentada de forma resumida a revisão da literatura realizada para esta pesquisa. Na Seção 4.1 é detalhado o protocolo da revisão da literatura e na Seção 4.2 são apresentadas as considerações finais. As informações completas sobre esta revisão da literatura podem ser obtidas no trabalho de Vaz [38].

4.1 Protocolo da Revisão da Literatura

Conforme descrito no Capítulo 3, o protocolo para a revisão da literatura visa estruturar de modo lógico todas as etapas necessárias para a realização de uma pesquisa formal, com o propósito de se obter um conjunto de produções científicas consistentes e relevantes para o estudo. Dentre as etapas necessárias para a realização da revisão da literatura é fundamental definir como será efetivamente executada a pesquisa, isto é, definir quais ações serão efetuadas em cada etapa da pesquisa. Como já mencionado anteriormente, a construção do protocolo de pesquisa utilizado neste estudo teve como base o modelo proposto por Kitchenham [21]. Assim, para esta revisão da literatura foram definidas as seguintes etapas: (i) Fase I – Planejamento, (ii) Fase II – Execução e (iii) Fase III – Resultados Obtidos, que são apresentadas a seguir.

4.1.1 Fase I – Planejamento

Dentre as diferentes ações realizadas para a construção do protocolo da revisão da literatura é imprescindível a clara definição do objetivo de pesquisa. Convém salientar que durante a pesquisa exploratória realizada inicialmente foram identificados poucos materiais que tratassem do tema da documentação das tarefas em Projetos de *Software Crowdsourcing*. Logo, a questão de pesquisa foi definida propositalmente de forma abrangente com o objetivo de minimizar possíveis restrições a artigos e viabilizar a correta identificação dos materiais científicos relacionados ao tema de pesquisa. Sendo assim, a questão de pesquisa foi definida conforme segue:

Como é realizado o processo de documentação das tarefas em projetos de *Software Crowdsourcing*?

Outras etapas relevantes no processo da revisão da literatura compreendem a definição das fontes de pesquisa e os termos utilizados nesta investigação. Para isso foram definidos como fontes de pesquisa os mecanismos mais frequentemente utilizados na Engenharia de Software, são eles: *IEEEExplore*, *ScienceDirect*, *Scopus*, *ACM* e *Springer*. Em relação aos termos de pesquisa, estes

tiveram como base a pesquisa de MAO [26], que apresenta uma revisão literatura que contempla diversos temas relacionados ao *Software Crowdsourcing*. Considerando este material, o conjunto de termos foi adaptado à necessidade desta pesquisa, o que resultou no seguinte grupo de termos: *software crowdsourcing*, *crowdsourcing software engineering*, *crowdsourcing software development*, *requirements*, *crowdsourcing requirements*, *requirements engineering*, *task requirements* e *task documentation*.

4.1.1.1 Artigos de Controle

Artigos de Controle são materiais previamente identificados que devem constar como resultado das buscas realizadas na revisão da literatura. São utilizados como uma das formas de validar a *string* de pesquisa. Neste estudo a identificação de artigos de controle foi realizada previamente pelo pesquisador através de uma pesquisa exploratória simplificada e teve como objetivo ampliar os conhecimentos sobre o tema de *Software Crowdsourcing*. Para esta pesquisa exploratória foi utilizado o método de *Snowballing*¹. Os artigos de controle utilizados nesta revisão da literatura são indicados a seguir.

Em seu artigo intitulado “*Crowdsourcing: A taxonomy and systematic mapping study*”, Hosseini [15] apresenta um estudo sobre *crowdsourcing* através da construção de uma taxonomia para o tema. O referido autor trata desta temática de forma ampla a fim de apresentar uma taxonomia aplicável a diversas propostas de estudo, uma vez que aborda características relacionadas aos elementos base do *crowdsourcing* (Contratante, Plataforma, Multidão e Tarefas). Dentre os principais pontos tratados neste artigo destacam-se para esta pesquisa as referências relacionadas às tarefas, visto que estas possuem relação direta com o tema desta investigação.

Mao [26, 27] desenvolve dois estudos que apresentam uma revisão da literatura sobre *crowdsourcing* de forma extensa, descrevendo diversos nichos de atuação pelo viés do desenvolvimento de software. O autor apresenta os diversos aspectos da engenharia de software aplicados a este contexto, além de explorar os elementos básicos do modelo consolidando informações relacionadas a eles. Como ponto de destaque nestes artigos pode-se apontar as questões relacionadas aos requisitos de software para o *Software Crowdsourcing*.

4.1.2 Fase II – Execução

A partir das definições propostas na Fase I foi realizado efetivamente o processo de revisão da literatura, no qual os termos de pesquisa foram estruturados conforme cada um dos motores de pesquisa. Este processo foi repetido diversas vezes até que se pôde finalmente chegar a um conjunto de termos adequados e que resultassem em materiais científicos relevantes para a pesquisa. De forma a ilustrar a evolução deste processo pode-se citar que na primeira execução da pesquisa foram

¹*Snowballing* é uma ferramenta de pesquisa que propõe a verificação de múltiplos trabalhos a partir das referências propostas no material de pesquisa inicial [41].



Categorias	Nº Artigos	% Artigos
Selecionados	40	42,6 %
Rejeitados	45	45,9 %
Duplicados	9	9,6 %
Total	94	100,0 %

Figura 4.1 – Consolidação da fase de avaliação de artigos da revisão da literatura

Fonte: Elaborado pelo autor

obtidos um total de 6796 artigos e ao final deste processo se obteve 94 artigos, que foram analisados integralmente e utilizados na revisão da literatura.

Assim, reforça-se que a partir das pesquisas realizadas e dos retornos obtidos em cada ciclo de pesquisa, os documentos foram sendo selecionados levando-se em consideração os que possuíam um maior alinhamento com o tema de pesquisa. Após o ciclo final de pesquisa nos motores de busca foram selecionados 94 artigos. Com base nesta seleção, foi realizada uma nova etapa de triagem com estes artigos a fim de verificar sua aderência ou não ao tema desta revisão da literatura. Como demonstrado na Figura 4.1, do total de 94 artigos selecionados inicialmente, 40 foram escolhidos por possuírem uma relação direta com o tema de pesquisa. Dos artigos restantes, 45 foram rejeitados por não apresentarem uma relação direta ao tema de pesquisa e nove são documentos duplicados obtidos das diferentes fontes de pesquisa.

4.1.3 Fase III – Resultados

Nesta seção são apresentados os resultados obtidos na revisão da literatura. Estes foram reunidos de conforme os critérios de avaliação utilizados na seleção dos artigos e são descritos na Tabela 4.1.

C1. Apresenta processos sobre a documentação das tarefas relacionados a *Software Crowdsourcing*?

A utilização deste critério tem como objetivo inicial identificar se a literatura descreve como as tarefas são documentadas e os respectivos processos utilizados para esta atividade. Como resultado deste critério de avaliação foram identificados dez artigos [33], [24], [12], [36], [34], [20], [9], [13], [39] e [8].

Este conjunto de artigos discute principalmente como ocorre a identificação e/ou seleção de participantes na multidão com o objetivo de mantê-los engajados como partes interessadas (*sta-*

Tabela 4.1 – Critérios de avaliação dos artigos selecionados

Critérios de Avaliação		Forma de Avaliação
C1	Apresenta processos sobre a documentação das tarefas relacionados a <i>Software Crowdsourcing</i> ?	Sim / Não
C2	Descreve técnicas tradicionais de engenharia de requisitos utilizados na documentação de tarefas em <i>Software Crowdsourcing</i> ?	Sim / Não
C3	Apresenta processo, métodos, técnicas ou ferramentas aplicadas à documentação das tarefas ou engenharia de requisitos em <i>Software Crowdsourcing</i> ?	Sim / Não
C4	Apresenta a descrição de qualidade de uma tarefa e sua documentação associadas a processos de <i>Software Crowdsourcing</i> ?	Sim / Não

Fonte: Elaborado pelo autor

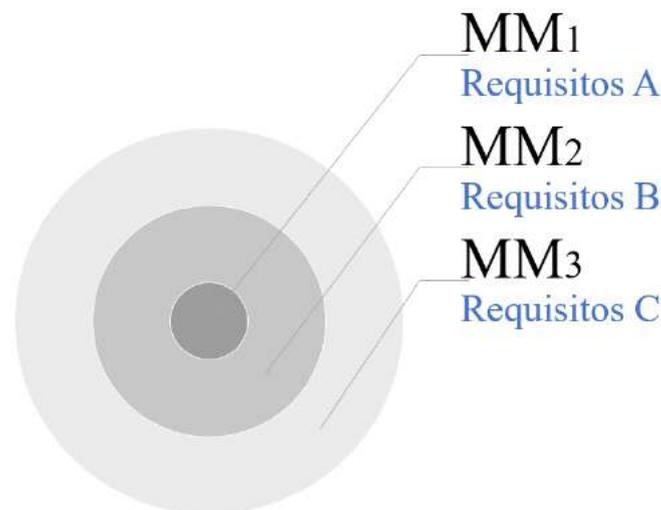


Figura 4.2 – Modelo evolutivo de Micro Multidões (MM) e dos requisitos obtidos

Fonte: Adaptado de [24]

keholders). Tendo em vista que a identificação das partes interessadas é um fator imprescindível para a Engenharia de Requisitos, este processo é fundamental como forma de construir uma rede coesa e focada em um tema específico, tornando possível a obtenção de um conjunto de requisitos consistentes.

Pode-se destacar o trabalho de Meira [24], que propõe um modelo para conceber uma multidão focada em problemas específicos e construída a partir de redes sociais, por exemplo: LinkedIn. A ideia proposta é evoluir estas micro multidões e a cada ciclo, quando necessário, agregar novas micro multidões, conforme ilustrado na Figura 4.2. Desta forma, é expandida a capacidade da rede tanto no seu conhecimento como na viabilidade de gerar novos requisitos.

Outra abordagem para a identificação de requisitos é apresentada no artigo de Groen [12], o qual propõe a construção de um ERP fictício que deve ser implantado em diversos locais da Europa. A demanda é obter os requisitos que atendam às particularidades de cada localidade onde será implantado o sistema. Com isso, é proposto utilizar a estratégia de *Software Crowdsourcing*

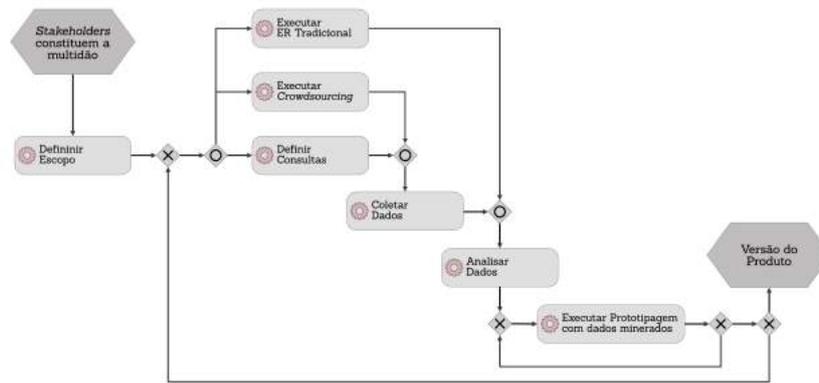


Figura 4.3 – Etapas do processo de seleção e documentação de requisitos [12]

Fonte: Adaptado de [12]

para identificar participantes em diversas localidades e com conhecimentos relevantes capazes de fornecer os requisitos necessários ao desenvolvimento do sistema. O fluxo das operações propostas pelo autor pode ser observado na Figura 4.3.

Como um contraponto aos artigos selecionados destaca-se o estudo apresentado por Dalpiaz [8], que descreve uma proposta para a elicitaco de requisitos funcionais do tipo ilimitados (*unbounded*) a partir de *Software Crowdsourcing*. Neste artigo o autor descreve este tipo de requisito funcional que, devido s suas caractersticas,  considerado complexo para documentaco por conta da variabilidade de informaoes necessrias para a sua validao. Segundo o referido autor, este tipo de requisito foi categorizado como: Inteligente, Subjetivo, No Generalizvel, Dinmico, Incerto e Impraticvel. Por exemplo, um requisito do tipo Subjetivo poderia ser a identificao de um local para eventos, j um requisito do tipo Incerto seria determinar se o clima poder afetar um evento. Nesta abordagem o autor considera que a utilizao do *Crowdsourcing* poder contribuir na elicitaco de requisitos em funo do conhecimento que a multido possui.

C2. Descreve tcnicas tradicionais de engenharia de requisitos utilizados na documentaco de tarefas em *Software Crowdsourcing*?

Com a utilizao deste critrio buscou-se verificar quais artigos tratam das tcnicas tradicionais de documentaco da tarefa ou engenharia de requisitos. Baseado neste critrio de avaliao foram identificados 2 artigos [12] e [36], que apresentam um estudo de caso semelhante cuja proposta  desenvolver um ERP (*Enterprise Resource Planning*) fictcio denominado MyERP utilizando uma plataforma de *Software Crowdsourcing*. Nos artigos citados existe a referncia aos modelos tradicionais da Engenharia de Requisitos sem, no entanto, detalhar tcnicas para a realizao da definio das tarefas para o ERP.

Em seu estudo, Groen [12] faz referncia  norma ISO/IEC/IEEE 29148 que trata da Engenharia de Requisitos. Esta norma define a ER como uma “funo interdisciplinar que intermedia os domnios do adquirente e do fornecedor para estabelecer e manter os requisitos a serem atendidos pelo sistema, software ou servio de interesse” [19]. Mesmo sem apresentar um detalhamento sobre ferramentas ou tcnicas utilizadas, Groen [12] apresenta um comparativo entre Engenharia de

Tabela 4.2 – Comparativo entre a ER Tradicional e a ER *Crowd-Based*

Aspectos da ER		ER "Tradicional"	ER <i>Crowd-Based</i>
Elicitação	Modo	Manual, Co-presencial	Semi-automatizado, remoto
	Técnicas	Entrevistas, Workshops, <i>Focus Group</i> , Observação, etc	Log de Arquivos e Análise de <i>Feedback</i> , Prototipação, Vídeo Conferência, etc
	Resultados	Verbalização dos Requisitos	Derivações <i>on-line</i> e Padrões de Uso
Documentação		Processamento Manual e/ou Assistido por Computador	Algoritmos semi-automatizados para filtragem, Derivações, Elaboração e Compilação (classificação, priorização, correlação, agrupamento)
Validação & Negociação		Co-Presencial	<i>Crowdsourced</i> , remoto
Gerenciamento		Iterativo dentro do projeto	Iterativo, Repetido em tempo livre
Perfil utilizado na ER		Engenheiro de Requisitos	Engenheiro de Sistema ou Requisitos
Tamanho Ideal		Depende do investimento e complexidade	Ilimitado (quanto mais Stakeholders, maior a validade do requisito)
Duração Total		Tipicamente muitos meses, em vários estágios	Análise recorrente de dados estáticos, Gerenciamento contínuo de multidões/comunidades

Fonte: Adaptado de [12]

Requisitos Tradicional e a Engenharia de Requisitos *Crowd-Based*, conforme demonstrado na Tabela 4.2.

Srivastava [36], em seu artigo, aborda o tema da elicitación dos requisitos com um foco diferenciado relacionado ao projeto MyERP, identificando e classificando os tipos de requisitos quando utilizados em *Software Crowdsourcing*. Neste estudo a identificação de técnicas tradicionais ocorre de forma superficial, apenas com citações de algumas técnicas na etapa de elicitación, como por exemplo: *workshop*, entrevistas, dentre outros.

C3. Apresenta processo, métodos, técnicas ou ferramentas aplicadas à documentação das tarefas ou engenharia de requisitos em *Software Crowdsourcing*?

Este critério foi utilizado para identificar quais artigos apresentam ou descrevem processos aplicados ao tema desta investigação. Para este critério de avaliação não foi feita nenhuma distinção em relação ao tipo de proposta dos autores (proposta teórica ou prática). Com base neste critério de avaliação foram identificados onze artigos que tratam do tema de pesquisa da revisão da literatura [33, 24, 45, 42, 12, 36, 34, 9, 13, 39, 8]. A abordagem dos autores é focada na identificação dos participantes e engajamento dos mesmos para que possam desempenhar o papel de *stakeholders*, com o objetivo de construir um conjunto consistente de requisitos necessários ao desenvolvimento de aplicações. Para obter estes resultados são apresentadas variadas estratégias, tais como: *frameworks*, modelos, plataformas, dentre outros.

Pode-se destacar o artigo de Snijders [34], que propõe o uso de uma plataforma para selecionar e engajar os participantes com o intuito de mantê-los produtivos na definição ou refinamento dos requisitos. Este objetivo de manter as partes engajadas e motivadas no projeto diz respeito a uma indicação do *Standish Group* apresentada no relatório *CHAOS Report* de 2009, que enfatiza o seguinte: "o sucesso de um projeto de tecnologia está diretamente relacionado com o envolvimento dos usuários". A abordagem de coleta de requisitos tem por base as histórias de usuários do modelo

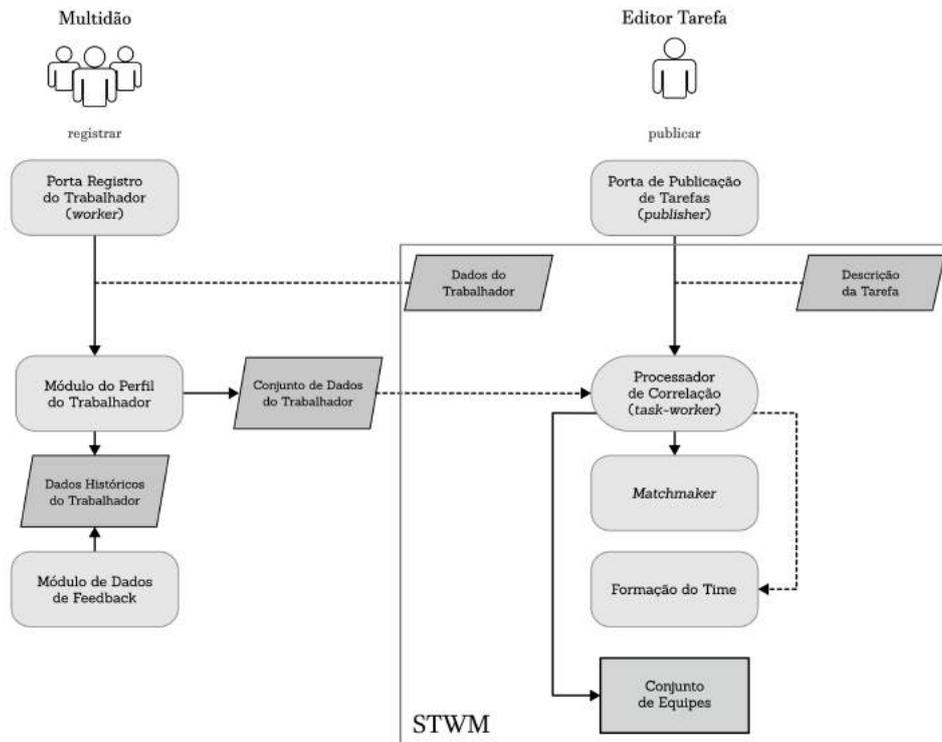


Figura 4.4 – Fluxo de processamento e validação de tarefas proposto por FU [9]

Fonte: Adaptado pelo autor de [9]

ágil de desenvolvimento de software. A plataforma proposta usa elementos de *gamification* para torná-la mais intuitiva aos participantes e fortalecer o envolvimento dos mesmos entre si e com a plataforma.

Em sua pesquisa Fu [9] propõe um meta-modelo cíclico que permite a verificação de possíveis recomendações de membros da multidão capazes de realizar a tarefa. A identificação dos possíveis candidatos é obtida a partir de características intrínsecas à documentação da tarefa bem como dos membros da multidão. O fluxo de operação proposto para a verificação de possíveis recomendações ocorre da seguinte forma: (i) se forem identificados membros na multidão capazes de realizar a tarefa o processo segue e a tarefa é publicada para a multidão, (ii) se nenhum participante da multidão for identificado para realizar a tarefa, esta será encaminhada para que sua descrição seja adequada e retorne novamente ao processo de verificação de possíveis candidatos. Este processo de verificação de possíveis recomendações permite à plataforma adequar o fluxo de trabalho interno visando o direcionamento das tarefas para uma melhor otimização de suas atividades. O fluxo descrito no artigo pode ser observado na Figura 4.4.

C4. Apresenta a descrição de qualidade de uma tarefa e sua documentação associadas a processos de *Software Crowdsourcing*

Este critério de avaliação teve como objetivo identificar características relacionadas à qualidade da documentação gerada pelo contratante e/ou plataforma e fornecida aos membros da

Tabela 4.3 – Modelo base para documentação de tarefas proposto por Shao [33]

Característica	Descrição	Tipo
Data Limite	Quanto tempo os desenvolvedores possuem para realizar a tarefa	Numérico
Recompensa	Quanto irá ganhar um desenvolvedor depois que ele for considerado o vencedor (será um intervalo de valores, média de valores)	Numérico
Validações Intermediárias	Quantas verificações de uma tarefa o contratante solicitará em média	Numérico
Plataforma	Plataformas que os desenvolvedores deverão utilizar, por exemplo: "Android", "Web"	Categórico
Linguagem	Linguagens que os desenvolvedores deverão utilizar, por exemplo: "Java", "Python"	Categórico
Tipo	Qual é o tipo da tarefa, por exemplo "Corrigir erros", "Desenvolver software"	Categórico
Descrição	Descrições detalhadas definidas pelo contratante, incluindo o título	Documento

Fonte: Adaptado pelo autor de [33]

multidão em projetos de *Software Crowdsourcing*. Do conjunto de quarenta artigos selecionados foram identificados 3 artigos [33], [12] e [36] que tratam deste tema.

Em seu artigo, Shao [33] propõe um *framework* para a recomendação de possíveis membros da multidão capazes de realizar uma tarefa. O *framework* define que a documentação deve possuir um conjunto padrão de informações que permita o seu processamento automatizado, conforme apresentado na Tabela 4.3. Com estes dados definidos para a tarefa é realizado o processamento destas informações através de redes neurais e análise de conteúdo utilizando padrões definidos pelo *framework*. Desta forma faz-se a recomendação dos participantes.

Os artigos de Groen [12] e Srivastava [36] abordam com perspectivas diferenciadas um mesmo tema, que é a utilização do conhecimento da multidão para a definição de requisitos de um ERP fictício. A descrição destes artigos é apresentada a seguir.

No artigo de Groen [12], é proposto um modelo em que uma multidão previamente selecionada através de uma rede de profissionais (*stakeholders*) fornecem os requisitos necessários para a construção de documentação adequada às necessidades do sistema. No fluxo proposto existem diferentes tarefas que vão auxiliar no refinamento e adequação da documentação, como por exemplo: prototipação com uso de mineração de dados, análise de dados, dentre outras atividades. Este fluxo pode ser observado na Figura 4.3.

Já o artigo de Srivastava [36] que também aborda um caso envolvendo um ERP fictício que visa obter os benefícios de uma base distribuída geograficamente e do conhecimento da multidão proposto pelo modelo de *Software Crowdsourcing* para a definição dos requisitos de sistema. Este estudo trata de alguns desafios inerentes ao modelo, como por exemplo: seleção de membros da multidão, identificação de tarefas adequadas que venham a captar as reais necessidades para aplicação, gestão de requisitos conflitantes e a priorização dos requisitos.

4.2 Considerações sobre a Revisão da Literatura

A partir desta revisão da literatura verificou-se que dentre o material obtido a maior incidência de registros trata da extração de requisitos através da multidão. Isto significa que é utilizada a capacidade da multidão de elaborar requisitos (seja através do conhecimento, disponibilidade ou localidade) de forma a atuar como parte interessada, isto é, a multidão pode atuar com diferentes perfis e informações para a construção dos requisitos. A partir da identificação dos requisitos, estes serão avaliados pelo contratante e conforme a sua definição vão compor ou não a documentação de novas tarefas. Em outras palavras, no contexto analisado a documentação se origina a partir de processos de coleta e/ou refinamento de requisitos obtidos pela multidão que serão adequados a uma nova etapa de *Software Crowdsourcing*: o desenvolvimento.

Considerando a questão de pesquisa, convém destacar o estudo de Shao [33] que apresenta explicitamente um padrão para a documentação da tarefa. Em sua abordagem é proposta uma adequação da documentação para agregar algumas informações relevantes, tais como: data limite para entrega, recompensa proposta, validações intermediárias envolvidas na tarefa, plataforma atribuída à tarefa, linguagem prevista para o desenvolvimento da atividade, tipo e descrição da tarefa. Com esta proposta o autor busca trazer uma maior clareza à documentação da tarefa e assim otimizar os processos por ele propostos. Os detalhes deste padrão de documentação foram apresentados anteriormente na Tabela 4.2.

5. ESTUDO DE CASO

Neste capítulo é apresentado de forma detalhada o planejamento, as ações e os resultados alcançados com a realização do Estudo de Caso. Como descrito na Seção 3.4 do Capítulo 3, com o Estudo de Caso é possível obter as percepções e opiniões dos participantes em um contexto de vida real. Para tanto, todas as etapas executadas durante este estudo são especificadas neste capítulo. Na Seção 5.1, são apresentados os sujeitos da pesquisa e sua caracterização. Na Seção 5.2, é apresentado o planejamento para a realização do Estudo de Caso e o contexto de sua execução. Na Seção 5.3, é detalhado o processo de coleta dos dados. Na Seção 5.4, são revelados os resultados do Estudo de Caso. Na Seção 5.5, é realizada a análise dos resultados obtidos e na Seção 5.6, são apresentadas as limitações identificadas durante a realização deste Estudo de Caso.

5.1 Sujeitos da Pesquisa

A identificação dos sujeitos da pesquisa é um fator essencial para que se possa fazer o planejamento da pesquisa, bem como adequar os instrumentos de coleta de dados ao público em questão. Neste sentido, a seleção do público-alvo para realizar o Estudo de Caso teve como base os alunos regularmente matriculados na disciplina de “Desenvolvimento Colaborativo de Software” do Programa de Pós-Graduação em Ciência da Computação da Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), no período letivo de 2017/2. A escolha desta disciplina e, consequentemente destes estudantes, deu-se em função do alinhamento do conteúdo e do plano de ensino da disciplina com o tema desta investigação. A turma selecionada era composta por 21 alunos.

A seguir são apresentadas as características dos participantes, obtidas a partir do questionário de Perfil dos Participantes realizado no Ciclo 1 do Estudo de Caso. Estas informações foram organizadas da seguinte forma: (i) formação, (ii) experiência em Tecnologia da Informação - TI e (iii) experiência em tecnologias. Estas informações podem ser observadas na Tabela 5.1 e cada um destes grupos é apresentado a seguir.

Formação

Os participantes deste Estudo de Caso possuem um conjunto diversificado de formações (cursos de graduação). Assim, pode-se agrupar os participantes conforme a sua formação: (i) Formação vinculada à TI, neste grupo existem 15 participantes cuja formação está diretamente relacionada à área de TI, os participantes que compõem este grupo são: P2, P3, P4, P6, P7, P8, P9, P10, P12, P13, P16, P17, P18, P19 e P20, (ii) Formação relativamente vinculada à TI, neste grupo há apenas um participante que possui formação em Administração com ênfase em TI, este participante é o P1, e (iii) Formação não vinculada à TI, neste grupo existem quatro participantes cuja formação não possui relação direta com a área de TI, são eles: P11, P14, P15 e P21.

Tabela 5.1 – Resumo dos dados de perfil dos participantes do Estudo de Caso

Participantes	Formação	Experiência					Experiência em Linguagens																		
		Tempo Total II	ANA	DES	DSG	TST	Angular	Android	Bootstrap	C	C++	CSS	Docker	Groovy	HTML5	Ionic	Java	JavaScript	JSON	JSP	MySQL	Node	Oracle 10g	Python	ReactJS
P1	Bacharelado Administração	15	12	15	0	10	8	D	D	E	C	E	E	C	E	C	E	C	C	C	C	E	C	D	E
P2	Tecnologia da Informação	20	10	10	0	20	0	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	D	E
P3	Sistemas de Informação	8	1	3	0	0	1	D	E	E	C	E	E	E	E	C	E	C	E	D	D	C	E	D	E
P4	Sistemas de Informação	8	0	2	0	0	0	B	E	C	D	E	E	D	E	B	D	E	E	E	B	E	E	E	E
P5	-	0	0	0	0	0	0	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
P6	Ciência da Computação	15	5	6	4	0	0	D	D	D	C	D	D	E	D	E	D	C	C	D	D	D	C	E	D
P7	Sistemas de Informação	5	1	5	0	1	1	E	C	D	C	E	B	C	D	A	E	B	A	A	D	C	A	E	C
P8	Sistema de Informação	13	10	13	1	6	1	E	D	D	B	D	C	E	B	E	A	A	A	B	A	D	A	D	A
P9	Ciência da Computação	15	4	15	2	1	1	E	E	D	E	C	E	E	E	E	C	E	D	E	C	E	D	E	E
P10	Bacharel em Informática	20	20	20	0	0	20	E	E	E	E	C	D	E	E	E	B	E	E	E	E	E	E	E	E
P11	Administração	20	20	5	0	5	0	C	E	E	E	C	E	E	C	E	C	E	C	E	E	C	E	C	E
P12	Sistemas de Informação	6	0	6	0	0	0	C	D	E	B	E	C	D	E	B	E	B	B	D	C	D	D	E	E
P13	Tec. em Processamento de Dados	17	0	0	0	2	14	E	E	E	E	C	E	E	D	E	C	E	E	E	C	E	E	E	E
P14	Comunicação Social	9	9	0	0	0	0	E	E	E	D	D	E	E	D	E	E	E	E	E	D	E	E	E	E
P15	Engenharia Mecânica	1	0	0	0	0	0	D	E	E	D	D	E	E	C	E	C	E	E	E	E	E	E	B	E
P16	Informática	22	10	0	0	0	0	E	E	E	E	B	D	E	E	D	E	E	E	E	E	E	E	C	E
P17	Sistemas de Informação	7	0	5	2	0	2	D	E	C	D	D	C	E	D	C	C	D	D	E	C	D	E	E	E
P18	Ciência da Computação	15	5	6	0	10	0	D	E	E	C	D	E	E	E	C	E	C	E	D	C	E	E	E	E
P19	Ciência da Computação	10	0	7	0	0	7	E	E	E	B	D	C	E	D	E	D	E	D	E	D	E	E	C	E
P20	Ciência da Computação	8	2	8	0	0	2	A	C	E	C	C	C	E	B	D	B	B	C	C	C	D	B	C	E
P21	Jornalismo	6	6	0	1	2	0	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E
Média		12	115	126	10	57	57																		
Total			115	126	10	57	57																		

Legenda Experiência Linguagens

- A: Tenho domínio avançado
 B: Conheço muito bem
 C: Sei utilizar
 D: Conheço pouco
 E: Não sei utilizar
 F: Não Aplicável

Legenda Experiência II

- ANA: Análise
 DES: Desenvolvimento
 DSG: Design
 GES: Gestão
 TST: Teste

Experiência em TI

Com relação à experiência em TI relatada pelos participantes, destaca-se a expressiva variação no que diz respeito ao tempo total de experiência em TI que cada um destes participantes possui, visto que há um participante com um ano de experiência e outro com 22 anos de experiência. Como apresentado na Tabela 5.1, observa-se que o tempo médio de experiência dos participantes é de 12 anos. Além de obter a informação relativa à experiência total em TI, foi solicitado aos participantes que detalhassem esta informação respondendo a questão relativa aos perfis recorrentes na área de TI (Análise, Desenvolvimento, *Design*, Gestão e Teste). Cabe evidenciar que muitos das respostas dos participantes apresentaram divergências entre o tempo total e o somatório de sua experiência profissional. Isso se deve ao fato de que alguns destes participantes relataram tempos de experiência em diferentes perfis que ocorreram de forma concomitante. Determinados participantes apresentaram variadas experiências na área de TI atuando com perfis diferentes dos sugeridos no questionário, conforme indicado a seguir:

- *“Gestão da Qualidade em Engenharia de SW (6 anos), Gestão do Conhecimento e Inovação (15 anos).” (P2)*
- *“Antes de trabalhar com testes, trabalhei como instrutora de cursos de TI e técnica em manutenção de HW e SW.” (P13)*
- *“3 anos de experiência em projeto de componentes mecânicos. 1 ano como auto-ditada em programação Java, Python e .Net.” (P15)*
- *“Suporte de TI na área de infraestrutura das empresas.” (P16)*
- *“Experiência em: Redes de Computadores, Servidores GNU/Linux, Segurança da Informação, Virtualização e Computação em Nuvem. Enfoque de trabalho sempre em backend.” (P19)*

É interessante observar as diferentes experiências que cada participante possui quando se analisa a formação e o tempo de experiência de cada um, o que torna o Estudo de Caso mais rico. Ainda sobre os dados apresentados, convém esclarecer que o participante P5 não respondeu ao questionário de perfil por ter ingressado na disciplina após o período em que este questionário estava disponível aos participantes.

Experiência em Tecnologias

Na Tabela 5.1 são estratificadas algumas informações relativas à experiência dos participantes com determinadas tecnologias. A seleção deste conjunto de tecnologias foi feita a partir da consulta às tarefas disponíveis na plataforma TopCoder nos últimos seis meses a contar da data de confecção do instrumento de coleta de dados. Foram consultadas as tarefas abertas e encerradas no período indicado e a seguir consolidado o conjunto de tecnologias presentes nestes tarefas, de modo a tornar o instrumento mais próximo da realidade. A coleta destes dados contou também

com uma questão textual na qual os participantes podiam apresentar observações em relação às suas experiências.

Considerando que o tempo médio de experiência dos participantes com TI é de 12 anos e que para alguns esse tempo pode ultrapassar os 20 anos, nota-se que há participantes que possuem experiência em linguagens mais antigas ou até mesmo atuam em segmentos não representados nos perfis disponíveis no formulário fornecido, como por exemplo infraestrutura. Pode-se destacar alguns dos relatos dos participantes em relação às suas experiências em tecnologias, conforme apresentado a seguir:

- *“Cobol, Fortran, Basic, Progress, Assembler...faz muito tempo!”* (P2)
- *“Assembly - Sei utilizar (depende do hardware) BASH - Sei utilizar”* (P3)
- *“Minha maior experiência é com a linguagem ABAP, linguagem proprietária da SAP.”* (P9)
- *“Já desenvolvi pequenos sistemas em Clipper, Delphi e atualmente trabalho com ADVPL (linguagem proprietária para o ERP Protheus, da TOTVS).”* (P11)
- *“Atuei como desenvolvedor somente na minha graduação. Que por sinal faz bastante tempo (finalizada em 1995).”* (P16)
- *“BASH: Domínio avançado Pascal: Conheço muito bem”* (P19)

5.2 Planejamento e Execução

Tendo em vista as características inerentes ao Estudo de Caso e as necessidades previstas para esta pesquisa, este Estudo de Caso foi planejado para ser realizado em dois ciclos. No Ciclo 1 pressupõe-se que os participantes são iniciantes em *Software Crowdsourcing* e na plataforma TopCoder e neste modelo de trabalho, logo estariam se ambientando com a forma de trabalhar na plataforma, manifestando as suas primeiras impressões. No Ciclo 2, acredita-se que estes participantes já entendem os mecanismos de funcionamento da plataforma e possuem uma relativa experiência em utilizá-la, assim suas percepções apresentam diferentes (e as vezes conflitantes) pontos de vista, o que permite ao pesquisador obter novas informações. Assim, o planejamento proposto para o Estudo de Caso pode ser observado na Tabela 5.2. Já o detalhamento de cada uma destas etapas é apresentado a seguir:

Detalhamento: Ciclo 1 do Estudo de Caso

- 1 Elaborar instrumentos de coleta de dados: foi realizada a definição e a construção dos instrumentos de coleta de dados utilizados no Ciclo 1. Todos os artefatos produzidos nesta etapa foram disponibilizados em meio eletrônico e são listados a seguir:

Tabela 5.2 – Plano de Ação do Estudo de Caso para o Ciclo 1 e Ciclo 2

Plano de Ações - Ciclo 1	Plano de Ações - Ciclo 2
1. Elaborar instrumentos de coleta de dados:	1. Adequar dos instrumentos de coleta de dados:
(a) Questionário - Perfil do Participante	(a) Questionário - Seleção da Tarefa
(b) Questionário - Seleção da Tarefa	(b) Questionário - Avaliação da Tarefa
(c) Diário de Bordo	2. Validar e Testar Questionários
(d) Questionário - Avaliação da Tarefa	3. Coletar dados - Seleção da Tarefa
2. Validar e Testar Questionários	4. Acompanhar execução da tarefa
3. Coletar dados - Perfil do Participante	5. Anotações de aula
4. Coletar dados - Seleção da Tarefa	6. Coletar dados - Avaliação da Tarefa
5. Acompanhar execução da tarefa	7. Consolidar dados coletados
6. Anotações de aula	8. Análise preliminar dos resultados
7. Coletar dados - Avaliação da Tarefa	9. Reunião de discussão com o grupo
8. Consolidar dados coletados	
9. Análise preliminar dos resultados	

Fonte: Elaborado pelo autor

- a Questionário - Perfil do Participante: este questionário foi criado com o objetivo de conhecer o perfil dos participantes do estudo. Estas informações permitem identificar e qualificar o tempo e o tipo de experiência que os participantes possuem no segmento de TI. Além disso, foi possível identificar qual o nível de conhecimento dos participantes sobre algumas das linguagens de programação mais utilizadas na plataforma TopCoder, dentre outras informações. Este formulário foi disponibilizado aos participantes através da plataforma Qualtrics¹. O modelo do formulário pode ser encontrado no APÊNDICE A.
- b Questionário - Seleção da Tarefa: a aplicação deste questionário teve como objetivo registrar o entendimento e a compreensão da tarefa selecionada pelo participante, assim como perceber as motivações que o fizeram selecionar uma tarefa em detrimento de outras. Este formulário foi disponibilizado aos participantes através da plataforma Qualtrics. O modelo do formulário pode ser encontrado no APÊNDICE B.
- c Diário de Bordo: este é um artefato produzido para que os participantes pudessem documentar o andamento das suas atividades relacionadas à resolução da tarefa. O documento possui uma estrutura aberta, ou seja, permite que o participante faça anotações e registre as atividades efetuadas durante o desenvolvimento da tarefa. Com esses dados, pretendeu-se avaliar as percepções dos participantes sobre a realização da tarefa, as dificuldades encontradas para a realização da mesma, dentre outros aspectos. Este artefato foi disponibilizado aos participantes através da plataforma Google Drive, em que cada participante possuía um arquivo exclusivo para realizar os seus registros. O modelo do formulário pode ser encontrado no APÊNDICE C.
- d Questionário - Avaliação da Tarefa: este questionário foi criado com a finalidade de obter os resultados e opiniões dos participantes sobre a execução da tarefa selecionada. Dentre as informações obtidas através deste instrumento estão os dados relacionados

¹Qualtrics: Plataforma de pesquisas *on-line* disponível aos alunos da PUCRS

a como os participantes perceberam a qualidade da documentação, como eles classificam a documentação fornecida, dentre outras informações. Cabe salientar que dentre as questões propostas neste questionário existe uma que solicita aos participantes que classifiquem a documentação da tarefa de acordo com um conjunto pré-definido de critérios. Estes critérios foram definidos a partir da consolidação das três propostas de classificação de requisitos apresentadas na Seção 2.4 do Capítulo 2 deste trabalho. Estes critérios foram extraídos de Wiegiers [40] e dos Guias BABOK 2 [17] e BABOK 3 [18] e consolidados de forma completa, isto é, todos os critérios de todas as três fontes foram considerados em um único formato e apresentados no formulário. Estes critérios foram reunidos de acordo com a sua origem: (i) Wiegiers (Preciso, Necessário), (ii) BABOK 2 (Coesão, Ajustabilidade/adaptação, Não Ambiguidade, Correção), BABOK 3 (Atômico, Completo, Consistente, Viabilidade, Conciso, Testável) e por fim um critério foi obtido a partir da compilação de Wiegiers e do BABOK3 que foi o critério de Priorizado. Este formulário foi disponibilizado aos participantes através da plataforma Qualtrics. O modelo do formulário pode ser encontrado no APÊNDICE D.

- 2 Validar e Realizar Pilotos dos Questionários: nesta etapa do processo os artefatos gerados foram avaliados e testados pelo autor, pela orientadora e por pesquisadores do Grupo MuND-DoS, que também estudam sobre o tema de *Software Crowdsourcing*. Esta etapa também contou com mais de um piloto dos instrumentos a fim de adequá-los para a sua utilização na pesquisa.
- 3 Coletar dados - Questionário - Perfil do Participante: com o instrumento de perfil validado, este questionário foi disponibilizado para todos os participantes.
- 4 Coletar dados - Questionário - Seleção da Tarefa: com o andamento da disciplina foram apresentados os conceitos básicos relacionados ao tema *Crowdsourcing* e *Software Crowdsourcing*. Posteriormente, foi apresentada a plataforma TopCoder, onde os participantes iriam realizar as atividades da disciplina. Neste contexto, cada participante iria avaliar as tarefas disponíveis e selecionaria apenas uma conforme sua preferência. Nessa etapa esse questionário já estava disponível aos participantes. As informações que envolviam os motivos da seleção ou descarte de tarefas foram obtidas a partir deste artefato.
- 5 Acompanhar a execução da tarefa: o acompanhamento das tarefas ocorreu de duas formas distintas: (i) durante toda a execução das tarefas o pesquisador e a orientadora respondiam as dúvidas apresentadas pelos participantes que envolviam conceitos sobre o tema, dúvidas relacionadas às tarefas, dentre outros assuntos e (ii) durante a execução da tarefa os participantes deveriam registrar no Diário de Bordo suas percepções relativas ao andamento da sua tarefa, problemas e soluções encontradas para resolução da tarefa, problemas com a plataforma, entre outros assuntos que entendessem pertinentes. Constantemente o pesquisador e a orientadora acompanhavam os relatos apresentados pelos participantes realizando questionamentos e solicitando esclarecimentos sobre as observações prestadas pelos participantes.

- 6 Anotações de aula: esta foi uma etapa que transcorreu durante todo o período em que a disciplina foi ministrada. Em diversos momentos foi possível observar como os participantes interagem com a plataforma e/ou relatavam suas percepções em relação à plataforma e às tarefas disponíveis na mesma.
- 7 Coletar dados - Questionário - Avaliação da Tarefa: nesta etapa os participantes já haviam realizado as suas tarefas, desse modo cada um pôde preencher o questionário. Esse formulário apresenta questões que envolvem a compreensão da tarefa, o entendimento por parte do participante sobre a qualidade da documentação em relação às necessidades da tarefa, problemas identificados na realização da tarefa e possíveis sugestões para melhorar a documentação das tarefas.
- 8 Consolidar dados coletados: ao final do Ciclo 1 todos os artefatos correspondentes a esta etapa foram coletados, organizados e consolidados.
- 9 Análise preliminar dos resultados: de posse dos dados foi possível avaliar os resultados obtidos no Ciclo 1, o que viabilizou a realização de ajustes nos instrumentos de coleta de dados para o Ciclo 2.

Com base na análise dos resultados preliminares do Ciclo 1, verificou-se a necessidade de realizar ajustes nos procedimentos utilizados, principalmente nos instrumentos de pesquisa a fim de adequá-los às necessidades do próximo ciclo. Todos os artefatos produzidos no Ciclo 2 foram disponibilizados em meio eletrônico. As atividades realizadas no Ciclo 2 podem ser observadas na Figura 5.2 e são detalhadas a seguir:

Detalhamento: Ciclo 2 do Estudo de Caso

- 1 Adequar os instrumentos de coleta de dados: com base nas informações coletadas no Ciclo 1, foi possível realizar ajustes nos artefatos com o intuito de deixá-los ainda mais adequados aos objetivos da pesquisa, além de torná-los mais claros. Os artefatos ajustados nesta etapa foram:
 - a Questionário - Seleção da Tarefa: foram realizados ajustes pontuais no texto de determinadas questões deste questionário. O objetivo deste formulário é o mesmo apresentado para o Ciclo 1. Este formulário foi disponibilizado aos participantes através da plataforma Qualtrics. O modelo com o formulário pode ser encontrado no APÊNDICE E.
 - b Questionário - Avaliação da Tarefa: foram realizados ajustes nos textos de determinadas questões deste formulário, assim como também foram acrescentadas questões complementares que podem fornecer mais detalhes acerca das impressões dos participantes em realizar as tarefas da plataforma TopCoder. Cabe destacar a alteração realizada nos critérios de classificação da documentação da tarefa. Nesta questão, os títulos de cada

categoria foram removidos e o total de critérios foi reduzido em função de alguns destes critérios não serem relevantes nesta avaliação ou mesmo por conta da impossibilidade dos participantes realizarem esta avaliação. Os critérios descritos a seguir foram retirados do questionário devido aos seguintes motivos: (i) Consistente: o critério solicita uma avaliação se a documentação esta alinhada com as necessidades das partes interessadas. Como os participantes não possuem contato com as partes interessadas torna-se inviável realizar este tipo de avaliação. (ii) Necessário: solicita verificar se a funcionalidade descrita na documentação agrega valor ao negócio. Este tipo de avaliação muitas vezes não é factível uma vez que o real objetivo de negócio previsto para a tarefa na plataforma em geral não é explicitado. (iii) Ajustabilidade/adaptação: solicita a verificação se as tarefas inter-relacionadas estão corretamente documentadas. Esta identificação pode ser difícil de se realizar por conta de não existir uma indicação objetiva da existência ou não deste tipo de tarefa na plataforma. e (iv) Priorizado: solicita a verificação se o requisito esta devidamente priorizado quando comparado aos demais requisitos da aplicação. Como não é viável identificar outras tarefas correlacionadas, a própria identificação da priorização fica prejudicada para sua avaliação. Este formulário foi disponibilizado aos participantes através da plataforma Qualtrics. O modelo com o formulário pode ser encontrado no APÊNDICE F.

- 2 Validar e Realizar Pilotos dos Questionários: da mesma forma como ocorreu no Ciclo 1, os artefatos gerados foram avaliados e testados pelo autor, pela orientadora e por pesquisadores do Grupo MuNDDoS, que também estudam sobre o tema de *Software Crowdsourcing*. Esta etapa também contou com mais de um piloto dos instrumentos a fim de adequá-los para a sua utilização na pesquisa.
- 3 Coletar dados - Questionário - Seleção da Tarefa: ao iniciar o Ciclo 2, os participantes selecionaram uma tarefa que melhor se ajustasse ao seu perfil e ao seu conhecimento, do mesmo modo que ocorreu no ciclo anterior. Após a seleção da tarefa, este questionário foi preenchido pelo participante. Os objetivos deste questionário são os mesmos apresentados para o Ciclo 1.
- 4 Acompanhar a execução da tarefa: seguiu os mesmos padrões e ações previstas para o Ciclo 1 desta pesquisa.
- 5 Anotações de aula: seguiu os mesmos padrões e ações previstas para o Ciclo 1 desta pesquisa.
- 6 Coletar dados - Questionário - Avaliação da Tarefa: seguiu os mesmos padrões e ações previstas para o Ciclo 1 desta pesquisa.
- 7 Consolidar dados coletados: seguiu os mesmos padrões e ações previstas para o Ciclo 1 desta pesquisa.
- 8 Análise preliminar dos resultados: seguiu os mesmos padrões e ações previstas para o Ciclo 1 desta pesquisa.

9 Reunião de discussão com o grupo: ao finalizar o Ciclo 2, foi realizada uma discussão com o grupo de participantes a respeito das suas experiências ao realizar as tarefas na plataforma TopCoder.

5.3 Coleta de Dados

Tendo em vista que o Estudo de Caso tem como sujeitos da pesquisa os alunos matriculados na disciplina de Desenvolvimento Colaborativo de Software, as ações efetivas para a coleta dos dados foram iniciadas após a realização de algumas aulas da disciplina. Considerando que o modelo de *Software Crowdsourcing* representa um modelo diferenciado de terceirização, optou-se por abordar inicialmente na disciplina os conceitos relacionados ao desenvolvimento distribuído de software. Após esta etapa, foram apresentados aos alunos os temas de *Crowdsourcing* e *Software Crowdsourcing*, este último que foi o foco do trabalho por eles realizado na disciplina. Concluída a apresentação dos conceitos, foi realizada uma demonstração para os alunos sobre as funcionalidades da plataforma TopCoder e os tipos de tarefas disponíveis para a seleção.

Quando do encerramento da apresentação da plataforma foi liberado aos participantes o primeiro questionário, que visa obter dados para a identificação do perfil de cada um deles. Este questionário foi liberado de forma eletrônica a todos os participantes. A coleta destes dados torna-se importante na medida em que se pretende compreender como a formação, a experiência, dentre outras questões pode influenciar na realização de tarefas na plataforma TopCoder.

Encerrada a etapa de coleta dos dados dos perfis dos participantes, foi efetivamente iniciado o processo de realização das tarefas pelos participantes. Esta etapa, como mencionado anteriormente, é composta por dois ciclos: (i) Ciclo 1 - etapa em que os participantes são novatos no uso da plataforma, bem como nos conceitos e regras utilizadas na mesma. A partir desse cenário, é esperado obter um conjunto de percepções mais voltados à utilização/operação da plataforma, e (ii) Ciclo 2 - nesta etapa os participantes já possuem um relativo conhecimento das regras e da plataforma. Assim, acredita-se que as percepções relatadas terão um foco diferente daqueles apresentados no Ciclo 1, devido à experiência anterior.

O processo de seleção das tarefas que cada um dos participantes iria realizar foi inteiramente livre, ou seja, não houve nenhuma interferência neste processo. Cada participante deveria acessar a plataforma TopCoder e avaliar, dentre as tarefas disponíveis, aquela que em seu entendimento ele fosse capaz de executar até o fim, isto é, submeter uma solução à plataforma. Ao finalizar este processo de seleção, os participantes deveriam responder a um questionário que solicitava a indicação da tarefa selecionada, bem como o motivo da seleção desta tarefa e o motivo que o levou a desconsiderar outras tarefas disponíveis. Com os resultados deste questionário pôde-se identificar quais critérios cada participante utilizou na seleção da sua tarefa e quais fatores foram relevantes para desconsiderar outras tarefas.

No decorrer dos dois ciclos do Estudo de Caso houve o constante acompanhamento dos participantes na realização das tarefas tanto presencialmente (durante o período de aula), como virtualmente, por meio de um Diário de Bordo (arquivos disponíveis no *Google Drive*). Cada participante possuía um arquivo exclusivo para cada ciclo do Estudo de Caso e o acesso a cada um destes arquivos era pessoal, ou seja, somente o Participante, Pesquisador e a Orientadora poderiam acessá-lo. Este acesso tinha como objetivo auxiliar os participantes em eventuais problemas que eles estivessem enfrentando ou para que o diário apresentasse um melhor detalhamento das informações. Este acompanhamento frequente através do Diário de Bordo teve como propósito estimular que os participantes apresentassem um conjunto de informações rico em detalhes. Entretanto, um fator crucial e constante durante todo o processo de acompanhamento realizado através do Diário de Bordo foi o cuidado para não direcionar ou induzir os participantes a apresentarem uma afirmação ou resposta que pudesse gerar algum tipo de viés em relação à pesquisa. A seguir são destacados alguns dos comentários feitos pelo pesquisador e pela orientadora.

- *“Sugestão de grafia para seu diário - Que tal criar uma “label” para indicar o dia que a atividade relatada ocorreu? Assim rapidamente se poderá identificar o seu progresso.”* (Orientadora)
- *“Você chegou a encaminhar algum tipo de dúvida mim? Não notei progresso. Você esta com alguma dificuldade?”* (Pesquisador)
- *“P3, qual seu entendimento aqui? Poderia explicar um pouco mais, por favor?”* (Orientadora)
- *“P4, como você chegou a esta proposta de solução? Teve que realizar alguma pesquisa sobre o tema?”* (Pesquisador)
- *“P21, como foi o processo para chegar até essa solução? Quais dificuldades você encontrou? Como você se sentiu ao conseguir resolver o problema?”* (Pesquisador)
- *“Você tem alguma previsão de confirmação do aceite da tarefa?”* (Pesquisador)
- *“Bem observado! Como você chegou a esta conclusão?”* (Orientadora)

Como atividade final do Ciclo 1 os participantes realizaram o preenchimento do formulário de avaliação da tarefa. Neste artefato foi solicitado que os participantes realizassem algumas avaliações em relação à documentação da tarefa de modo a classificá-la. A partir destes resultados foi possível identificar aspectos em relação ao entendimento da tarefa que os participantes realizaram.

Para a realização do Ciclo 2 do Estudo de Caso foi necessário fazer alguns ajustes nos questionários de seleção da tarefa e avaliação da tarefa. Estes ajustes foram decorrentes da avaliação dos resultados parciais obtidos no Ciclo 1, em que observou-se que algumas questões não apresentavam informações adequadas ao objetivo da pesquisa. Os demais artefatos e procedimentos realizados no Ciclo 1 foram repetidos no Ciclo 2, de forma a se obter uma consistência no método de pesquisa aplicado no Estudo de Caso.

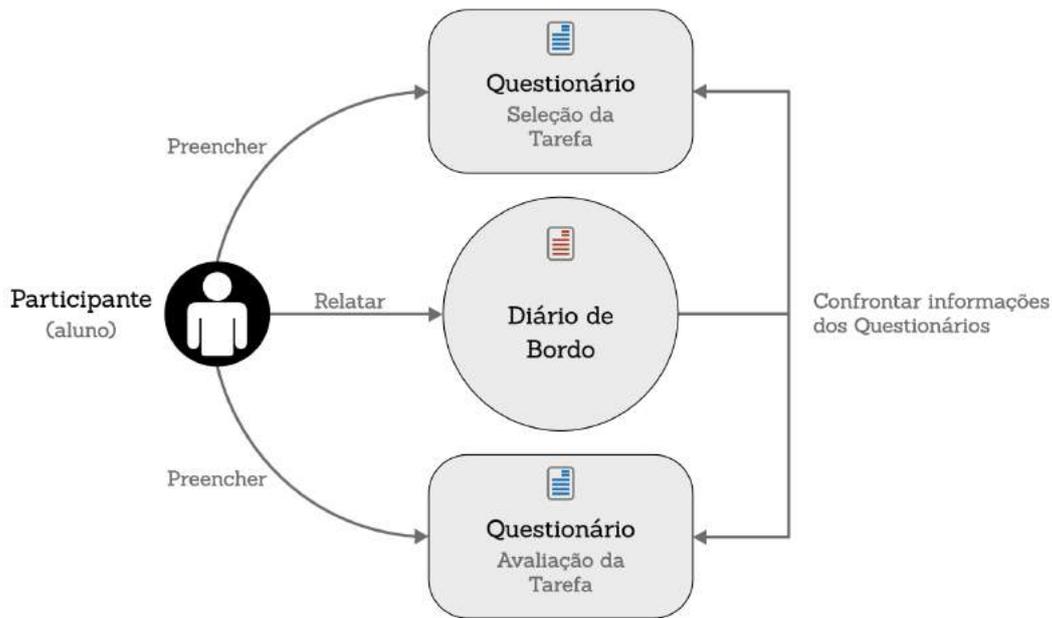


Figura 5.1 – Processo de avaliação das informações (Questionários e Diário de Bordo)

Fonte: Elaborado pelo autor

5.4 Resultados do Estudo de Caso

Nesta seção são apresentados os resultados obtidos a partir das informações coletadas durante a realização dos Ciclos 1 e 2 do Estudo de Caso. Estes resultados estão relacionados ao Objetivo 1 desta pesquisa. Cabe destacar que neste processo de consolidação e análise dos resultados a estrutura de informações utilizada teve como base a representação da Figura 5.1, ou seja, os participantes responderam aos questionários de cada ciclo (Seleção da Tarefa e Avaliação da Tarefa), bem como realizaram o relato detalhado de suas ações através do Diário de Bordo. Sendo assim, as informações prestadas nos formulários foram confrontadas com aquelas apresentadas nos diários para que desta forma fosse possível obter resultados mais precisos sobre o tema investigado.

Cabe reforçar que este Estudo de Caso procurou investigar as percepções que os participantes apresentam em relação à documentação das tarefas. Acredita-se que este processo de identificação da tarefa que o participante irá realizar envolve diversos aspectos e que estes possuem graus diferenciados de impacto para cada participante. Pode-se citar dentre estes fatores: o entendimento da tarefa, o prazo de entrega, as ferramentas necessárias para o desenvolvimento da tarefa, a tecnologia envolvida na tarefa, a premiação atribuída à tarefa, dentre outros. É importante esclarecer que estes diferentes fatores não são neste momento foco desta pesquisa.

Após finalizada a coleta de dados nos Ciclos 1 e 2 do Estudo de Caso, foi realizada a leitura exhaustiva do material e todas as informações foram consolidadas, analisadas e classificadas conforme os preceitos defendidos por Bardin [2]. Assim, são apresentados a seguir os principais achados deste estudo, que tratam da percepção dos participantes sobre a documentação das tarefas na plataforma TopCoder.

Quando se avalia os resultados obtidos através do formulário de seleção das tarefas, pode-se perceber que os participantes relatam dificuldades em compreender o que é efetivamente solicitado na tarefa, ou seja, não conseguem compreender a descrição da tarefa ou seus entregáveis. Logo, esta tarefa passa a ser desconsiderada. Se este processo é repetido diversas vezes, esta tarefa não será realizada no prazo esperado pelo Contratante, gerando prejuízos aos envolvidos. A seguir são destacados alguns dos relatos apresentados pelos participantes em relação à documentação das tarefas.

- *“Descrições confusas.
Tarefas que parecessem exigir mais de 10 horas para desenvolvimento/realização.
Não estava interessada em atividades onde precisasse fazer uma grande preparação prévia de ambiente em tecnologias que não iriam ser aplicáveis na minha pesquisa ou no meu trabalho a curto prazo.”* (P1)
- *“Achei as orientações muito confusas. Tive que ler e reler várias vezes o enunciado. Em especial a parte “Full Description & Project Guide” que não ficou claro onde terminava a descrição do que eu deveria considerar como conteúdo e o que eram requisitos do projeto que eu precisava entregar. Eles precisavam ter identificado melhor cada trecho.”* (P1)
- *“Entrando em alguns destes desafios consegui verificar o nível de informação oferecida pelo solicitante em relação ao objeto a ser desenvolvido é variada. Em alguns casos muito sucinta e em outros casos muito farta.”* (P2)
- *“O que mais me chamou a atenção na tarefa foram as keywords enumeradas. Elas não dizem absolutamente nada sobre como o logo deve ser. Life-changing, Driven, Tech-Savvy. Palavras de efeito mas de significado vazio.”* (P6)
- *“Tecnologia. Tempo para iniciar a execução. e Entendimento do problema.”* (P10)
- *“Entre as atividades que estavam abertas para registro, esta me pareceu mais acessível para entender na Topcoder.”* (P10)
- *“Iniciei com uma tarefa de design mas acabei perdendo o prazo de postagem da primeira rodada e, então, selecionei este novo desafio.
Desconsidereí outras tarefas por:*
 - *Julgar muito complexas;*
 - *Julgar que demandariam muito tempo;*
 - *Por não conhecer a tecnologia utilizada e por ser mais complexa;*
 - *Por não entender da regra de negócio da qual se tratava o desafio;”* (P11)
- *“Não consigo encontrar o que deve ser feito em lugar algum. Não há informações no fórum. Além disso não consigo acesso ao repositório da tarefa.”* (P14)

- *“Tive dificuldades de entender como deveria ser a submissão. Me pareceu mal explicado o challenge” (P15)*
- *“Foi fornecido um documento de requisitos funcionais com telas. Procurei pela telas e não as encontrei.” (P20)*

Da mesma forma em que foram identificados em diversos relatos que os participantes descrevem problemas com a documentação da tarefa no momento de sua seleção, pode-se identificar relatos semelhantes durante a execução das tarefas. Este tipo de registro é decorrente da natureza do desenvolvimento de software, em que somente durante a execução das tarefas que os participantes percebem a necessidade de algum detalhamento mais específico sobre a atividade. Estes relatos que ocorreram durante a execução das tarefas foram registrados no Diário de Bordo. A seguir são apresentados alguns fragmentos destes relatos extraídos dos diários:

- *“...não encontrei uma opção para imprimir a especificação da atividade, nem dos termos e condições de uso da TopCoder para facilitar salvar o documento...” (P1)*
- *“...na descrição do challenge é mencionado que tem além das questões iniciais tem um documento publicado no fórum que descreve algumas áreas para serem mensuradas. Porém, não encontrei o link para o Fórum mesmo fazendo o registro...” (P1)*
- *“...entrando em alguns destes desafios consegui verificar o nível de informação oferecida pelo solicitante em relação ao objeto a ser desenvolvido é variada, Em alguns casos muito sucinta e em outros casos muito farta...” (P2)*
- *“...Tentei olhar o fórum para ver se tinha alguma dica, mas não tinha nada...” (P7)*
- *“...Não tem mockup na descrição, não sei se entendi bem...” (P7)*
- *“...Encontrei algumas dificuldades em fazer a configuração do ambiente (Node.js) mais especificamente com problemas de versão...” (P8)*
- *“...É uma tarefa um pouco mais complexa do que eu gostaria e com tecnologias que não conheço muito, por isso não sei se vou conseguir terminá-la. Descobri que preciso me registrar no <http://predix.topcoder.com/>, mas a parte de registro do site não está funcionando...” (P9)*
- *“...Estou trabalhando no challenge e gostaria de mais informações sobre ele. Fui na parte de discussões do challenge e achei um post com perguntas muito pertinentes ao challenge, mas não achei satisfatória as respostas para algumas delas (e como são as mesmas perguntas que eu tenho, não adianta perguntar de novo). Por exemplo não fica explícito quem vai escanear do passaporte. Eles dizem que isso será feito em algum momento mas não dizem se algum funcionário ou se a própria pessoa vai fazer esse scan...” (P9)*

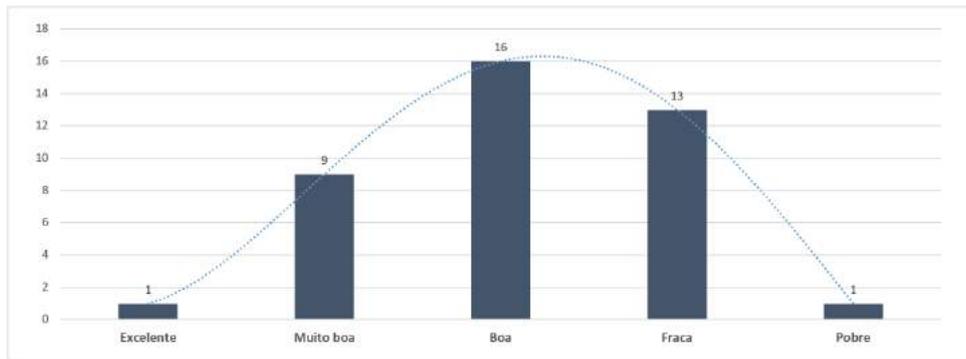


Figura 5.2 – Qualificação da documentação no Estudo de Caso - Ciclo 1 e 2

Fonte: Elaborado pelo autor

Em relação aos resultados obtidos a partir do formulário de avaliação da tarefa, é possível observar um conjunto de informações complementares que dizem respeito a como os participantes classificam a documentação das tarefas frente a alguns parâmetros pré-definidos. Estas informações são relatadas a partir dos seguintes questionamentos.

Como foi qualificada a documentação da tarefa realizada pelos participantes?

Os resultados encontrados a partir da coleta de dados realizada no Ciclo 1 e no Ciclo 2 podem ser observados no gráfico da Figura 5.2. Nesta figura é possível observar que a categoria de maior incidência em relação a qualidade da documentação encontra-se como “Boa”. Entretanto, ao analisar a curva de tendência apresentada no gráfico revela que as categorias “Fraca” e “Pobre” se destacam. O que demonstra que mesmo os participantes podendo realizar a escolha das tarefas, muitas ainda apresentam problemas em relação a documentação e assim prejudicando a realização da mesma. Dentre os relatos que demonstram estes fatos pode-se apresentar alguns segmentos extraídos do diários de bordo em que se pode constatar as dificuldades enfrentadas pelos participantes.

- *“Achei as orientações muito confusas. Tive que ler e reler várias vezes o enunciado. Em especial a parte “Full Description & Project Guide” que não ficou claro onde terminava a descrição do que eu deveria considerar como conteúdo e o que eram requisitos do projeto que eu precisava entregar. Eles precisavam ter identificado melhor cada trecho.” (P1)*
- *“O desafio é acompanhado de uma farta documentação mostrando como a loja funciona atualmente e o quais os objetivos a alcançar com o novo design...” (P2)*
- *“A tarefa era de fácil entendimento. Muito bem documentada e com um passo a passo muito bem escrito.” (P8)*
- *“Encontrei muita dificuldade quanto a explicações/informações no site do TopCoder, referentes a como proceder na resolução da tarefa e também a forma de envio da solução. Para informações sobre como proceder com a tarefa eu fui redirecionado ao site da Ethereum onde*

havia um tutorial com um passo a passo de como criar o ambiente e como proceder com a criação do Hello World. Porém no tutorial não mencionava a necessidade de fazer o download da carteira do Ethereum. Resumindo faltavam detalhes da tarefa tanto no site da TopCoder como no do Ethereum.” (P8)

- *“Muita documentação sobre o que precisa ser feito, mas o setup do ambiente não é claro. Há indicação passo a passo da codificação e vários trechos de códigos que podem ser utilizados. Além disso, no grupo de discussão há uma longa thread indicando dúvidas relacionadas ao entregável (um vídeo).” (P10)*
- *“Tive dificuldades de entender como deveria ser a submissão. Me pareceu mal explicado o challenge...” (P15)*

Como os participantes classificaram a documentação da tarefa realizada?

O objetivo dessa questão, proposta nos dois ciclos realizados neste Estudo de Caso, é obter as impressões dos participantes acerca da documentação da tarefa utilizando um conjunto de critérios pré-definidos e classificados através da Escala Likert². Dessa forma, cada participante pôde avaliar como a documentação da tarefa que ele realizou está ou não aderente a estes critérios. Como indicado anteriormente, a construção destes critérios teve como base três fontes que propõem classificações para os requisitos. De modo a auxiliar no acompanhamento e análise dos resultados deste item os critérios de classificação utilizados pelos participantes no Ciclo 2 são detalhados na Tabela 5.3.

O resultado consolidado da classificação da documentação da tarefa realizada pelos participantes pode ser observada na Tabela 5.4. Esta tabela apresenta os resultados obtidos no Ciclo 1 e 2 do Estudo de Caso. Assim, ao analisar o resultado consolidado, percebe-se que a maior incidência de respostas ocorre para as classificações “Concordo totalmente (CT)” e “Concordo parcialmente (CP)” em todos os critérios, o que demonstra que os critérios aplicados à Engenharia de Requisitos para classificação e qualificação são aderentes à documentação das tarefas em *Software Crowdsourcing*.

Durante o processo de identificação e classificação das tarefas foi solicitado aos participantes que indicassem como os critérios estariam alinhados ou não à documentação da tarefas que eles haviam realizado. Assim, é possível apresentar alguns fragmentos da documentação da tarefa fornecida pelos participantes com as respectivas avaliações como pode ser observado na Figura 5.3 e Figura 5.4.

²Escala Likert, é uma proposta de classificação de respostas que permite definir o nível de concordância do respondente frente a um tema.

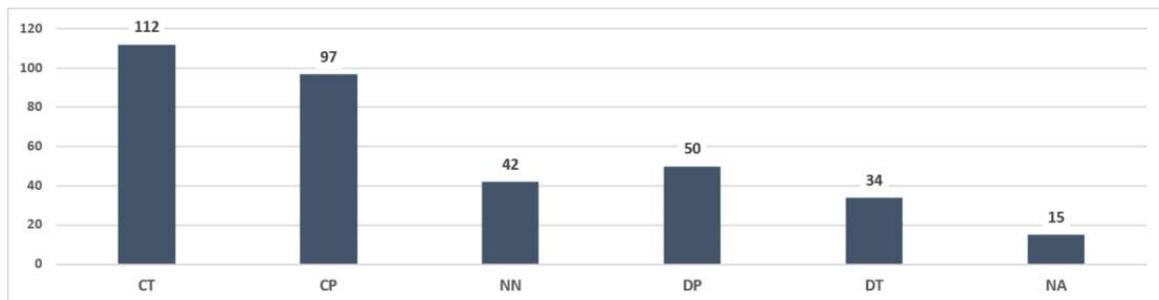
Tabela 5.3 – Listagem com os critérios de classificação da documentação da tarefa - Ciclo 2

ID	Descrição
C1	A documentação fornecida para a sua tarefa estava devidamente autocontida , ou seja, não necessita de outras informações para executar a tarefa.
C2	A documentação fornecida para a sua tarefa possui o nível de detalhamento necessário para executar a tarefa, ou seja, descreve as etapas que devem ser realizadas de forma satisfatória para a finalização da tarefa.
C3	A documentação fornecida para a sua tarefa possui o nível de precisão necessária para executar a tarefa, ou seja, descreve os detalhes de como os procedimentos devem ser realizados para finalizar a tarefa.
C4	A documentação fornecida para a sua tarefa está corretamente documentada e coesa de forma a dar o suporte necessário para executar a tarefa.
C5	A documentação fornecida para a sua tarefa descreve uma tarefa factível dentro do prazo e orçamento proposto.
C6	A documentação fornecida para a sua tarefa não possui informações desnecessárias para a execução da tarefa.
C7	A documentação fornecida para a sua tarefa possui informações dúbias , ou seja, pode apresentar interpretações divergentes conforme o interlocutor que estiver lendo esta documentação.
C8	A documentação fornecida para a sua tarefa possui informações suficientes para realizar os testes necessários a fim de validar se a tarefa foi cumprida conforme esperado.
C9	A documentação fornecida para a sua tarefa possui problemas que podem gerar defeitos no resultado final da tarefa.

Fonte: Elaborado pelo autor

Tabela 5.4 – Classificação da documentação da tarefa no Estudo de Caso - Ciclo 1 e 2

Critérios	Concordo totalmente (CT)	Concordo parcialmente (CP)	Não concordo nem discordo (NN)	Discordo parcialmente (DP)	Discordo totalmente (DT)	Não se aplica (NA)
C1	15	10	3	9	3	0
C2	11	12	6	9	2	0
C3	11	11	7	6	4	1
C4	11	11	8	7	2	1
C5	20	9	3	6	2	0
C6	20	12	3	3	2	0
C7	14	10	3	5	8	0
C8	8	10	6	5	3	8
C9	12	12	3	0	8	5
TOTAL	122	97	42	50	34	15



Fonte: Elaborado pelo autor

Quais pontos podem ser melhorados ou mantidos em relação à descrição da documentação das tarefas?

Todas as sugestões de melhorias ou aspectos a serem mantidos relatados pelos participantes do Estudo de Caso foram analisados, organizados e categorizados de forma rigorosa a fim de se obter um referencial relevante para esta pesquisa. O resultado consolidado destas informações pode

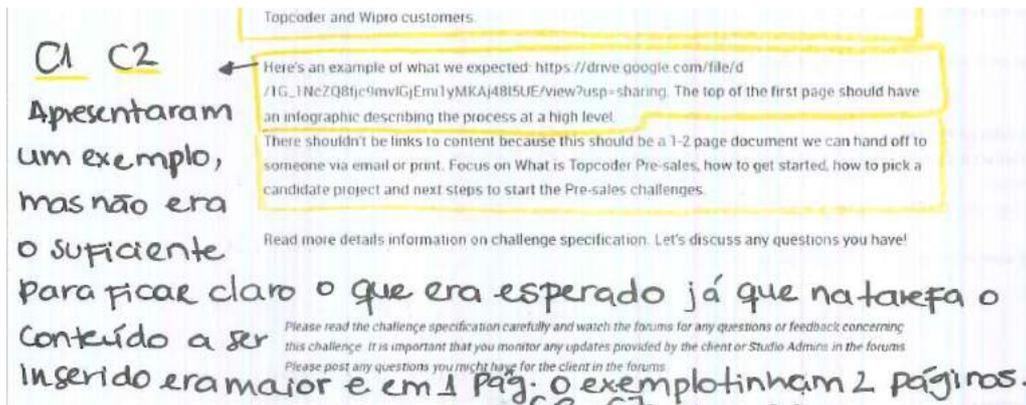


Figura 5.3 – Fragmento da avaliação da documentação da tarefa no Estudo de Caso (P1)

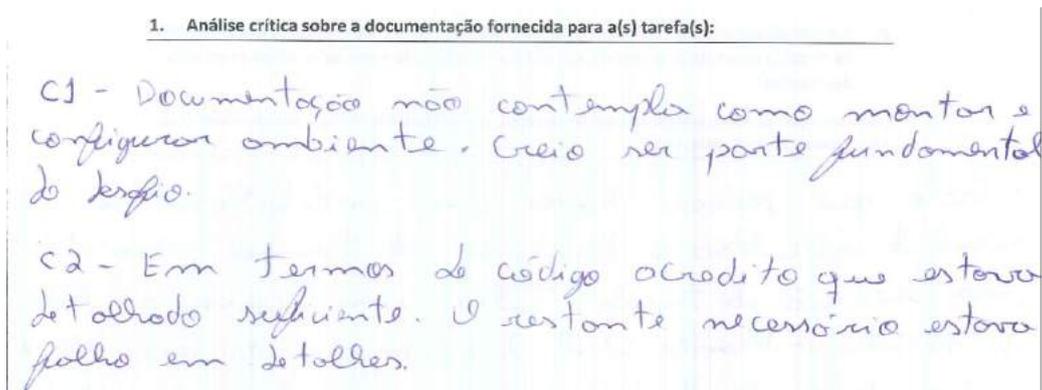


Figura 5.4 – Fragmento da avaliação da documentação da tarefa no Estudo de Caso (P11)

ser observado na Tabela 5.5, que apresenta a categorização obtida através do mapeamento das sugestões propostas pelos participantes. A identificação de classes utilizadas neste processo teve como base o método de análise de conteúdo proposto por Bardin. A identificação das classes foi realizada de forma incremental. Assim, de forma recorrente houve o refinamento das categorias até que se chegou ao conjunto apresentado. A tabela exhibe de forma unificada as informações relativas ao Ciclo 1, Ciclo 2 e também as informações obtidas na reunião de retrospectiva realizada com os participantes. Cada uma das sugestões propostas pelos participantes pode ter originado mais do que uma categoria, isso pode ser observado em diferentes momentos em que um mesmo participante é indicado em mais de uma categoria no mesmo ciclo. As categorias propostas na Tabela 5.5 são apresentados a seguir.

Conforme mencionado, o processo de identificação destas categorias ocorreu de forma evolutiva. Desse modo, ao realizar a análise da documentação completa foi possível observar em diversos momentos ocorrências em que os participantes relataram a necessidade de **Padronização**, o que poderia auxiliar tanto na organização das informações, como também no entendimento da tarefa, uma vez que estas se tornariam mais objetivas através de “blocos” de informações pré-definidas, deixando mais eficiente a identificação de informações. Dentre os relatos que tratam desta categoria, pode-se destacar o relato de P14, que cita um problema de uniformidade da apresentação das informações “...cada pessoa que cadastra uma tarefa tem a sua percepção do que deve ser

Tabela 5.5 – Classificação das sugestões do Estudo de Caso

Ciclo 1		Ciclo 2		Retrospectiva	
Classes do Modelo	Participante	Classes do Modelo	Participante	Classes do Modelo	Participante
Padronização		Padronização	P14, P19	Padronização	P7, P18, P19, P20
Resumo da Tarefa	P3, P9	Resumo da Tarefa		Resumo da Tarefa	
Plataformas		Plataformas		Plataformas	
Tecnologias		Tecnologias		Tecnologias	P3
Links Relacionados	P7, P11, P16	Links Relacionados	P8	Links Relacionados	P8, P3
Configuração do Ambiente		Configuração do Ambiente	P7, P11, P13, P15, P20	Configuração do Ambiente	P7, P10, P11, P13, P15, P18
Pré-Requisitos		Pré-Requisitos		Pré-Requisitos	P2, P8
Tarefas Relacionadas		Tarefas Relacionadas		Tarefas Relacionadas	P3, P7
Ferramentas	P20, P21	Ferramentas	P17	Ferramentas	P8, P11, P17
Detalhamento da Tarefa	P4, P9, P12, P13, P17, P20	Detalhamento da Tarefa	P1, P3, P4, P5, P6, P8, P10, P15, P16, P18, P20	Detalhamento da Tarefa	P1, P3, P5, P6, P9, P11, P13, P15, P16, P17, P20
Documentação Complementar	P1, P7, P11, P16	Documentação Complementar	P8, P17	Documentação Complementar	P7, P14
Crítérios de Teste e Aceitação	P4, P10, P17	Crítérios de Teste e Aceitação	P9, P12, P13	Crítérios de Teste e Aceitação	P3, P7, P9, P17, P18, P20
Regras de Submissão da Tarefa	P15, P17	Regras de Submissão da Tarefa		Regras de Submissão da Tarefa	P7, P9, P10, P17

Fonte: Elaborado pelo autor

colocado no enunciado ou não. O que deixa as tarefas, talvez fora de um standard...”. Este mesmo caso pode ser observado na indicação de P19 “*Padronização na documentação de todas as tarefas do TopCoder através de um checklist...*”

Em relação à categoria de **Resumo da Tarefa**, os participantes relataram que as informações apresentadas não foram suficientemente claras para transmitir o objetivo geral da tarefa. Devido a esta falha na construção desta parte da documentação, os participantes mencionaram problemas, como indicado por P3 “*...um sumário da tarefa na própria interface do TopCoder ajudaria, pois é necessário muita boa vontade para entender o que a tarefa propõe com apenas o título e as informações da página inicial do TopCoder...*” e P9 “*...documentação inicial foi bem fraca e faltaram muitas informações...*”. Além da própria identificação deste resumo da tarefa é importante destacar que a forma como estas informações são apresentadas faz uma grande diferença na compreensão da tarefa.

A categoria **Plataformas** permanece na apresentação da classificação das respostas dos participantes desse estudo como forma de manter uma padronização entre o Estudo de Caso e o Estudo de Campo e principalmente com o modelo proposto. Como demonstrado na Tabela 5.5 esta categoria não apresentou nenhuma ocorrência.

A categoria **Tecnologias**, que trata de descrever as tecnologias envolvidas na realização da tarefa, é frequentemente apresentada quando envolve tarefas de *design*. Entretanto, em outros tipos de tarefas essa informação não é apresentada, como por exemplo: *CODE* ou *First2Finish*. Esta indicação pode ser observada no relato de P3 que cita “*...deveria haver uma categorização mais detalhada das tarefas (exemplo design | Photoshop ou Code | Banco de Dados | MySQL | Triggers...*”.

Para a categoria de **Links Relacionados**, além dos relatos registrados através dos instrumentos utilizados neste Estudo de Caso, os participantes comentavam sobre a necessidade de concentrar as informações de links relacionados com a tarefa em um único local, a fim de simplificar a sua localização ao invés de revisar a documentação completa para encontrá-los, quando registrados. A partir desta perspectiva, pode-se destacar algumas das declarações dos participantes,

que tratam por exemplo do uso de mídias ou gráficos (links externos à plataforma) para auxiliar na compreensão das tarefas. Estes casos podem ser observados nas citações de P7 “... poderia ajudar um vídeo ou algo assim de um dos stakeholders comentando sobre o contexto e a necessidade daquela implementação...” ou de P11 “...poderiam haver vídeos explicativos ou mesmo diagramas para melhorar a compreensão. Creio que elementos gráficos e audiovisuais facilitam a compreensão...”.

Quando da realização das tarefas os participantes deste Estudo de Caso comentavam com frequência sobre problemas em relação à montagem do ambiente para desenvolvimento e/ou casos em que ao iniciar uma tarefa identificaram que a mesma necessitava de uma configuração que iria prejudicar outros ambientes pré-existentes na sua máquina, o que os levou a desistir de realizar a tarefa. Este tipo de relato foi categorizado como **Configuração do Ambiente**. Alguns destes relatos podem ser observados a seguir: P7 “...nem todos repositórios de código tem um documento de README com instruções para setup do ambiente, o que às vezes dificulta iniciar o desafio...”, P10 “...Poderia ter um idem de ambiente de desenvolvimento, indicando de forma mais clara qual o ambiente necessário (com API, BD, etc)...”, P11 “...Documentação complementar para auxiliar na montagem do ambiente (IDE, bibliotecas, etc)...” e que inclusive sugere a separação na documentação exclusiva para a configuração, P13 “...Melhorar os detalhes técnicos do setup de ambiente (talvez criar uma sessão separada para isso)...”.

Na medida em que os participantes realizavam as tarefas propostas na plataforma TopCoder perceberam que as mesmas apresentavam características específicas, isto é, necessitavam que os participantes possuíssem algum tipo de conhecimento sobre o tema envolvido com a tarefa. Esta identificação de conhecimentos prévios foi categorizada como **Pré-Requisitos**. Os participantes a seguir fazem comentários a respeito deste ponto: P2 “...acredito que para a seleção das tarefas poderia ser indicadas as características ou requisitos de conhecimento necessários à sua execução...” e P8 “...muitas vezes o pré-requisito da tarefa poderia ser mais detalhado...”.

Outra questão relatada durante a execução das tarefas no Estudo da Caso trata de correlação entre tarefas. Em alguns casos esta relação existente entre as tarefas pode ser percebida apenas em função de títulos semelhantes. Entretanto, em outros casos existe a informação de que a tarefa é sequência de outras, sem no entanto indicar quais são estas tarefas. A apresentação das tarefas relacionadas poderia auxiliar na compreensão total do que deve ser realizado em uma tarefa. Com esse propósito foi sugerida a categoria de **Tarefas Relacionadas**. Esses relatos foram apresentados por vários participantes, mas há dois casos que foram explicitados nos instrumentos de coleta, que são: P3 “...algumas tarefas pareciam ser continuidade de outras, mas a identificação é somente pelo padrão do título. Poderia haver um agrupamento...” e P7 “...nas tarefas que tem uma continuação, por exemplo, em que na tarefa 1 se cria um serviço e na tarefa 2 se adiciona campos na resposta do serviço, seria legal ter na tarefa 2 o link para as tarefas anteriores até para ter contexto...”.

Através das declarações dos participantes, constatou-se que em diversas tarefas propostas pela plataforma TopCoder existe a necessidade de utilizar ferramentas específicas, seja para o desenvolvimento ou para a entrega da solução da tarefa. Pensando nessa necessidade, foi proposta a categoria de **Ferramentas**, com o objetivo de concentrar estas informações em um só local,

ao contrário do que ocorre atualmente em que estas informações podem estar subentendidas ou dispersas em diferentes pontos da documentação. Alguns dos relatos dos participantes sobre este problema envolvendo as ferramentas necessárias à execução da tarefa estão destacados a seguir: P8 “...pode acontecer de faltar algumas informações como por exemplo a configuração do ambiente ou a necessidade de alguma ferramenta...”, P17 “...detalhar mais sobre essa ferramenta...”, P20 “...A documentação do front-end da aplicação era muito fraca, sendo esse front-end composto de inúmeros frameworks e ferramentas. Atualmente, devido a quantidade e variedade enorme de frameworks para front-end, é difícil que desenvolver...” e por fim P21 “...Poderiam deixar mais claras as opções de ferramentas necessárias para a execução por exemplo...”.

No momento em que um participante acessa uma tarefa o elemento mais relevante para a sua seleção é compreender o objetivo da tarefa. Para tanto, foi proposta a categoria de **Detalhamento da Tarefa**, que pretende concentrar a informação específica do que a tarefa deve realizar. Neste sentido, foram apresentados diversos relatos relacionados à compreensão da tarefa, como pode-se observar a seguir: P1 “...Mais clareza e organização na descrição da tarefa. Principalmente porque ela continha informações relacionadas aos processos do TopCoder então fiquei confusa se todo o conteúdo...”, P3 “...Para mim a maior dificuldade desta tarefa (e de outras) é identificar o que está sendo solicitado...”, P4 “...Descreverem melhor os requisitos e a forma em que será feita a avaliação e escolha, assim como o teste da qualidade da solução...”, P6 “...A tarefa em questão se utilizou de um linguajar extremamente vago, em que muitas coisas abstratas eram pedidas...”, P9 “...A documentação inicial foi bem fraca e faltaram muitas informações...”, P10 “...Melhorar a documentação, pensando em usuários mais experientes e menos experientes...” e P20 “...Poderia detalhar melhor os casos de uso e contextualizar a utilização do sistema...” dentre outros relatos.

Do mesmo modo em que o detalhamento da tarefa é relevante para a compreensão de uma tarefa, em muitos casos existe a necessidade de compor a documentação com outras informações que vão trazer mais detalhes para o participante. Estas informações, em geral, são apresentadas como anexos ou links disponíveis no fórum da tarefa. Devido às indicações dos participantes foi apresentada uma categoria específica que trata exclusivamente deste problema, que é **Documentação Complementar**. Alguns destes relatos são a seguir apresentados: P1 “...a documentação complementar poderia ter trazido mais detalhes do contexto da empresa...”, P16 “...A documentação deve estar em anexo já no momento da escolha da tarefa ao invés de termos de enviar um e-mail solicitando a mesma...”, P17 “...fornecendo exemplos de projetos que o software modela, e suas aplicabilidades também...” e P11 “...acredito que poderiam haver vídeos explicativos ou mesmo diagramas para melhorar a compreensão...”.

As duas últimas categorias possuem uma relação direta com a qualificação da entrega. Uma delas trata de como deve ser verificado se a tarefa foi executada conforme esperado, que é a categoria **Critérios de Teste e Aceitação**. Relacionada a este item, há a identificação de como uma tarefa deve ser submetida. Em tarefas que envolvem *design* é mais frequente que se tenha esta indicação, porém nas demais não há esta informação. Assim, foi identificada a categoria de **Regras de Submissão da Tarefa**, que propõe uma padronização, uma vez que permite que

esta informação esteja disponível para todos os tipos de tarefas. Para estas duas categorias pode-se apresentar os relatos dos participantes conforme segue: *“Descreverem melhor os requisitos e a forma em que será feita a avaliação e escolha, assim como o teste da qualidade da solução.”* P4, *“Nesta atividade, pouco poderia ser feito. Acredito que um checklist para indicar que a tarefa foi terminada corretamente poderia auxiliar.”* P10 e *“Pode ser melhor descritos os termos de aceite para a tarefa com os principais pontos que devem ser atendidos.”* P12 que apresentam pontos mais relacionados aos Critérios de Teste e Aceitação. Já em relação as Regras de Submissão da Tarefa alguns dos relatos dos participantes são: *“Acho que faltou mais clareza na forma de submissão das tarefas. A tarefa em si estava suficientemente clara, mas as questões que envolviam a própria plataforma poderiam ter sido melhor explicadas.”* P15, *“A questão do tempo de desenvolvimento, quando existe mais de uma entrega, para que possamos entender até quando e como funcionam esses "deadlines". Como a atividade era de design, faltou dizer se podia ou não utilizar o logo da competição (que no caso, não podia), mas eu usei e fiz a primeira entrega de um produto errado. E por fim, acho que faltou explicar o que era necessário conter nos arquivos entregáveis, já que era necessário explicar sobre o design do produto.”* P17, *“Diferentemente da tarefa 1 que realizei, essa tarefa estava bem documentada. Desta forma, achei a documentação satisfatória e não achei necessidade de pontos a melhorar na documentação. Talvez uma melhor descrição dos arquivos para entrega. E o feedback do checkpoint poderia ter sido mais rápido e detalhado.”* P9 e por fim *“Melhorar os detalhes técnicos do setup de ambiente (talvez criar uma sessão separada para isso). Especificar melhor as tarefas a serem entregues (especificar com mais clareza o que deve ser entregue).”* P13.

As informações completas com todas as sugestões são apresentadas no APÊNDICE I desta pesquisa.

5.5 Discussão sobre o Estudo de Caso

Ao realizar a retrospectiva dos aspectos tratados neste Estudo de Caso é importante reforçar que o foco desta pesquisa está relacionado à percepção que os participantes possuem sobre a documentação das tarefas. Assim, cabe salientar o que Wieggers [40] aborda sobre a especificação de requisitos, isto é, a documentação do requisito.

“A especificação de requisitos envolve a representação e armazenagem do conhecimento obtido dos requisitos coletados de forma persistente e bem organizada, ou seja, deve traduzir as necessidades relatadas pelos usuários e registradas em requisitos e diagramas escritos adequadamente para a compreensão, revisão e uso pelo público pretendido.”

Adaptado de [40] (p.17)

Neste sentido, constata-se que dentre os resultados obtidos em relação à seleção e descarte de uma tarefa pelos participantes, o aspecto referente à clareza da documentação desponta como um critério influenciador, porém não se destaca inicialmente como um fator explícito de motivação para a seleção de uma tarefa ou mesmo para desconsiderá-la. Este critérios surge a partir de iterações

Tabela 5.6 – Experiência em linguagens avaliadas por categorias no Estudo de Caso

Categories	.NET (ASP, C#)	Angular	Android	Bootstrap	C	C++	CSS	Docker	Groovy	HTML5	Ionic	Java	JavaScript	JSON	JSP	MySQL	Node	Oracle 10g	Python	ReactJS	TOTAL
Tenho domínio avançado	1	0	0	0	0	0	0	0	0	1	0	1	2	2	0	1	1	1	0	2	12
Conheço muito bem	1	0	0	2	2	0	1	0	0	3	0	5	2	2	1	1	0	1	1	0	22
Sei utilizar	2	2	2	3	8	5	7	3	0	3	1	8	3	2	2	10	0	3	4	0	68
Conheço pouco	6	4	5	2	5	10	6	2	1	7	1	2	3	3	5	3	5	3	4	1	78
Não sei utilizar	10	14	13	13	5	5	6	15	19	6	18	4	10	11	12	5	14	12	11	17	220
Não Aplicável	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	20

Fonte: Elaborado pelo autor

realizadas com os participantes e de seus relatos durante a execução da tarefa. Desta forma, percebe-se que em um primeiro momento ao realizar a seleção de uma tarefa, os participantes apresentavam um entendimento parcial de muitas destas tarefas. Entretanto, quando foram efetivamente realizá-las muitas dúvidas surgiram, algumas foram resolvidas através do fórum da plataforma, outras através de pesquisas externas à plataforma e algumas não foram solucionadas.

Outro fator que cabe destacar está relacionado ao grau de conhecimento que os participantes apresentavam em relação às tecnologias necessárias para realizar as tarefas. Considerando que a plataforma TopCoder utiliza-se de um modelo de negócio moderno, em geral as tarefas disponibilizadas possuem um elevado grau de inovação e, conseqüentemente, utilizam novas tecnologias. Observa-se ainda que o grupo de participantes deste Estudo de Caso, em sua maioria, apresentava um relativo grau de experiência em TI, mas baixo conhecimento nas tecnologias utilizadas nas tarefas disponibilizadas na plataforma, estas informações são apresentadas na Tabela 5.6. Este fator foi significativo no momento da seleção e execução das tarefas, uma vez restringe o volume de tarefas disponíveis para os participantes.

5.6 Limitações do Estudo de Caso

Neste Estudo de Caso foi possível realizar as atividades de pesquisa com um grupo de 21 pessoas, porém cabe mencionar que nem sempre todos os alunos da disciplina puderam executar todas as tarefas. Com base neste grupo, constatou-se que a diversidade na formação de cada um dos participantes e suas experiências prévias tornaram os resultados mais ricos. Entretanto, uma limitação significativa observada foi a de não ser possível realizar o Estudo de Caso acompanhando as tarefas em mais ciclos e com novos participantes cujos conhecimentos de tecnologias estivessem mais alinhados com as tarefas disponíveis na plataforma. Com a entrada de novos participantes e com novas etapas acredita-se que seria possível validar questões identificadas que envolviam o uso da plataforma (por exemplo: experiência no uso da plataforma, conhecimento das tecnologias envolvidas nas tarefas, dentre outras) e o entendimento da documentação da tarefa.

6. ESTUDO DE CAMPO

Neste capítulo é apresentado de forma detalhada o planejamento, as ações e os resultados obtidos com a realização do Estudo de Campo. Como descrito na Seção 3.5 do Capítulo 3, o Estudo de Campo visa obter as percepções e opiniões dos participantes buscando retratar a realidade. Sendo assim, este capítulo detalha as etapas executadas durante todo o estudo. Na Seção 6.1, são apresentados os sujeitos da pesquisa e sua caracterização. Na Seção 6.2, é apresentado o planejamento e a execução do Estudo de Campo. Na Seção 6.3, é detalhado o processo de coleta dos dados. Na Seção 6.4, são revelados os resultados do Estudo de Campo. Na Seção 6.5, é realizada análise dos resultados e na Seção 6.6, são apresentadas as limitações identificadas na realização deste Estudo de Campo.

6.1 Sujeitos da Pesquisa

Assim como no Estudo de Caso, no Estudo de Campo a identificação do sujeitos da pesquisa é fundamental para o desenvolvimento da investigação. Sendo assim, compreender as características necessárias para a escolha dos participantes torna-se um fator importante para a sua seleção. A partir desta perspectiva, dez profissionais com experiência em desenvolvimento WEB foram selecionados para participar deste Estudo de Campo, deste grupo, sete responderam ao questionário. O processo de seleção destes profissionais baseou-se na facilidade de acesso a eles, uma vez que todos trabalham na mesma empresas que o pesquisador. A premissa para a seleção destes profissionais foi possuir conhecimento nas linguagens Java e/ou Angular.

Considerando que os profissionais selecionados para este estudo possuem experiência nas tecnologias exigidas para avaliar as tarefas, a Tabela 6.1 apresenta o tempo total de experiência na área de TI que cada participante possui, assim como também as relaciona a alguns perfis mais recorrentes nesta área de atuação. No grupo investigado, o tempo médio de experiência na área de TI é de 11 anos. Com relação à experiência nas linguagens principais exigidas para a resolução das tarefas em avaliação pelos participantes, todos possuem nível sênior em Java e todos conhecem Angular em níveis variados como pode ser observado na tabela anteriormente citada.

6.2 Planejamento e Execução

Tendo em vista as características próprias ao Estudo de Campo definidas por McGrath [29] e as necessidades desta pesquisa, este estudo foi planejado para ser aplicado a um conjunto de profissionais capazes de realizar as avaliações pertinentes à documentação das tarefas propostas pela plataforma TopCoder. Assim, os questionários utilizados no Estudo de Caso foram adequados ao público-alvo deste Estudo de Campo, porém a sua estrutura em relação às informações coletadas foi mantida.

Tabela 6.1 – Experiência em TI dos participantes do Estudo de Campo

Participante	Experiência						Experiência Linguagens	
	Tempo Total TI	ANA	DES	DSG	GES	TST	JAVA	ANGULAR
P22	15	4	13	2	2	-	Sênior	Sênior
P23	9	-	9	-	-	-	Sênior	Sênior
P24	12	8	9	9	6	6	Sênior	Júnior
P25	18	-	18	10	-	-	Sênior	Júnior
P26	8	1	7	-	-	-	Sênior	Pleno
P27	5	-	5	-	-	-	Sênior	Pleno
P28	10	-	10	-	-	-	Sênior	Pleno
Média	11							
Total		13	71	21	8	6		

Legenda

ANA: Análise
DES: Desenvolvimento
DSG: Design
GES: Gestão
TST: Teste

Fonte: Elaborado pelo autor

Tabela 6.2 – Plano de Ação do Estudo de Campo

Plano de Ações
1. Adequar questionários do Estudo de Caso (Ciclo 1 e Ciclo 2):
(a) Questionário - Avaliação de Tarefas - Profissionais
2. Validar Questionários
3. Teste Piloto do Questionário
4. Coletar dados - Avaliação de Tarefas - Profissionais
5. Consolidar dados coletados
6. Análise preliminar dos resultados

Fonte: Elaborado pelo autor

O planejamento proposto para o Estudo de Campo pode ser observado de forma resumida na Tabela 6.2, o detalhamento de cada uma destas etapas está descrito a seguir:

- 1 Adequar questionários do Estudo de Caso (Ciclo 1 e Ciclo 2): para a construção e definição do instrumento que foi aplicado neste Estudo de Campo utilizou-se como base os formulários construídos para o Estudo de Caso (Ciclo 1 e Ciclo 2). Para adequar os formulários ao Estudo de Campo foram considerados tanto o público pretendido para participar do estudo como a dinâmica que seria proposta ao grupo. O artefato construído para este Estudo de Campo foi:
 - a Avaliação de Tarefas - Profissionais: para o Estudo de Campo o formulário de perfil foi suprimido, uma vez que já se possuía as informações básicas relativas à experiência dos participantes. Outras questões que foram removidas dizem respeito à experiência em utilizar a plataforma TopCoder, já que neste Estudo de Campo o foco foi totalmente

direcionado para a compreensão da documentação das tarefas. Desta forma, todas as questões foram direcionadas a este objetivo. Com isso, pretendeu-se que os participantes ao avaliarem as tarefas pudessem indicar, segundo sua compreensão, se a descrição destas tarefas apresentam as condições necessárias para que eles pudessem realizar a tarefa ou se existiam problemas vinculados a estas informações. O modelo com formulário pode ser encontrado no APÊNDICE H.

- 2 Validar Questionários: nesta etapa do processo o questionário foi avaliado e testado pelo pesquisador, orientador e por determinados pesquisadores do Grupo MuNDDoS, que também investigam sobre a temática de *Software Crowdsourcing*. Esta etapa contou com mais de uma rodada de avaliações e ajustes no instrumento a fim de adequá-lo para a sua disponibilização no estudo.
- 3 Teste Piloto do Questionário: tendo em vista que este questionário seria aplicado a profissionais com experiência em desenvolvimento de sistemas, foi adicionada uma fase após a validação para que 2 profissionais com experiências semelhantes a dos respondentes pudessem avaliar o formulário e apresentar suas considerações em relação aos itens: (i) perguntas utilizadas no formulário e (ii) forma de apresentação do problema aos participantes do Estudo de Campo. Suas impressões e sugestões foram pertinentes e contribuíram consideravelmente ao processo de ajuste do formulário, principalmente no que diz respeito ao processo de apresentação de contexto aos participantes.
- 4 Coletar dados - Avaliação de Tarefas - Profissionais: nesta etapa o pesquisador fez uma apresentação para os participantes do estudo com o propósito de conversar sobre o tema de pesquisa, explicar o estudo em curso e também apresentar o instrumento de pesquisa. Ao final desta apresentação os participantes receberam um material impresso contendo tarefas pré-selecionadas e o questionário para que pudessem realizar a avaliação do material e responder o questionário.
- 5 Consolidar dados coletados: ao final do prazo estipulado, os participantes que responderam o questionário tiveram seus dados armazenadas para análise posterior.
- 6 Análise preliminar dos resultados: de posse dos dados foi possível realizar a análise dos resultados a partir das informações prestadas pelos participantes do Estudo de Campo.

6.3 Coleta de Dados

Nesta seção são apresentadas as ações executadas para a realização da coleta dos dados do Estudo de Campo. Primeiramente foi definido o escopo que se pretendia investigar, que é obter a percepção dos participantes a respeito da documentação das tarefas disponibilizadas na plataforma TopCoder. Partindo dessa premissa foram selecionados 7 profissionais. O detalhamento das ações realizadas será apresentado conforme as etapas do planejamento anteriormente citadas.

1 Adequar questionários do Estudo de Caso (Ciclo 1 e Ciclo 2):

Nesta etapa do processo todos os formulários construídos para os dois ciclos do Estudo de Caso (Perfil do Participante, Seleção da Tarefa e Avaliação da Tarefa) foram novamente avaliados e estruturados em um único artefato. Este, por sua vez, foi adequado levando em consideração as características específicas deste Estudo de Campo, que são: (i) a experiência dos participantes já era de conhecimento do pesquisador por possuir contato profissional com os mesmos, portanto era desnecessário realizar determinado questionamento feitos no Estudo de Caso, (ii) os questionamentos realizados no Estudo de Caso relacionados a *Software Crowdsourcing* foram descartados visto que os participantes teriam uma apresentação sobre este assunto antes de realizar o preenchimento do questionário do Estudo de Campo, e (iii) tendo em vista que os profissionais possuem conhecimento nas linguagens Java e Angular e que os mesmos não teriam tempo hábil para realizar a seleção das tarefas na própria plataforma TopCoder, o artefato proposto para estes participantes contempla um conjunto de nove tarefas da linguagem Java e seis tarefas da linguagem Angular pré-selecionadas pelo pesquisador. A seleção destas tarefas se deu a partir da revisão das tarefas disponibilizadas nos últimos seis meses na plataforma a contar da data de confecção do instrumento de coleta. Estas tarefas deveriam possuir seus fóruns ainda abertos para que os participantes pudessem acessar eventualmente informações complementares existentes nestes locais. Convém salientar que na descrição de praticamente todas as tarefas propostas na plataforma TopCoder há a indicação de linguagens e/ou tecnologias envolvidas, portanto para a pré-seleção das tarefas que seriam avaliadas pelo grupo buscou-se a designação das linguagens Java e Angular.

a Avaliação de Tarefas - Profissionais:

Com a adequação deste artefato os participantes não precisaram acessar a plataforma TopCoder para selecionar as tarefas, uma vez que foram oferecidas tarefas pré-selecionadas relacionadas às linguagens Java e Angular. Esta estratégia teve como objetivo minimizar problemas de acesso à plataforma TopCoder, além de mantê-los focados em tarefas específicas. Assim, os participantes teriam que escolher algumas destas tarefas pré-selecionadas e, para cada uma delas, indicar qual ação iria realizar, podendo ser: (a) Lido, ou seja, o participante leu a descrição da tarefa e documentos correlatos que estivessem disponíveis sobre a tarefa e (b) Selecionado, indica que após a leitura desta tarefa o participante optou por realizar a tarefa. Outras adequação realizada no artefato diz respeito à questão que trata dos critérios de avaliação da documentação das tarefas. Nesta questão os participantes iriam apresentar suas percepções relacionadas ao conjunto das tarefas que selecionaram para realizar, indicando os critérios que considerassem positivos ou negativos relacionados à documentação. As demais questões apresentadas neste formulário seguem os padrões propostos no Estudo de Caso. Todo o material necessário para a análise e execução deste questionário foi impresso, com isso, além das questões, todas as tarefas sugeridas neste artefato foram devidamente extraídas da plataforma TopCoder e impressas como parte do material entregue aos participantes. Foi também

fornecido aos participantes um usuário e senha genérico na plataforma TopCoder para acessar as tarefas, se necessário. Juntamente com o usuário e senha foi informado o link de cada uma destas tarefas.

O processo de apresentação da pesquisa, bem como da entrega do questionário e tarefas pré-selecionadas ocorreu na empresa em que os participantes trabalham. Os participantes foram divididos em três grupos. Para cada um destes grupos foi realizada uma apresentação de aproximadamente meia hora sobre o tema de pesquisa (*Software Crowdsourcing*), o escopo da pesquisa e foram apresentadas as tarefas selecionadas. Em todos os grupos os participantes fizeram uma leitura rápida do questionário. As dúvidas que foram identificadas neste momento foram esclarecidas e os participantes finalizaram a avaliação posteriormente.

2 Validar Questionários:

Conforme descrito anteriormente, o artefato foi avaliado por pesquisadores com conhecimento no tema, propiciando assim, a definição de um questionário adequado ao público-alvo da pesquisa.

3 Teste Piloto do Questionário:

Após a validação interna do artefato, o mesmo foi apresentado a dois profissionais com experiência semelhante a dos participantes para que estes realizassem a avaliação do instrumento, bem como da apresentação da pesquisa (tema, foco de pesquisa, dentre outros detalhes). Os testes foram realizados individualmente com cada um destes profissionais. A estratégia neste caso foi obter as impressões destes profissionais sem nenhuma interferência relacionada ao conhecimento deste sobre o tema ou em relação ao artefato. Em cada um destes pilotos foram apresentadas sugestões para aprimorar o questionário, como por exemplo: apresentar exemplos que possam auxiliar no preenchimento de uma questão, destacar algumas palavras-chave de uma questão mais complexa para fortalecer a ideia que a questão pretende trabalhar. Já em relação à apresentação foram sugeridos detalhes na identificação de exemplos sobre *Crowdsourcing*, *Software Crowdsourcing* e na ordem de apresentação dos temas.

4 Coletar dados - Avaliação de Tarefas - Profissionais:

O processo de coleta dos dados deste Estudo de Campo foi executado em duas etapas: (i) a partir de uma apresentação sobre o tema de pesquisa, no qual foi exposto o conceito de *Crowdsourcing*, seu uso no ambiente atual, depois foi tratado do seu desdobramento em relação à *Software Crowdsourcing* e posteriormente tratado do tema relativo à requisitos e documentação das tarefas, e (ii) após a apresentação o autor repassou cada uma das questões do formulário com os participantes indicando o proposta da questão. Durante todo o processo, as dúvidas apresentadas foram esclarecidas, tanto aquelas relacionadas ao tema de pesquisa, como as que tinham relação com o formulário. Tendo em vista que o grupo de participantes foi heterogêneo e envolvia profissionais de diferentes equipes de trabalho, a alocação destes profissionais para uma atividade extra teve de ser segmentada, logo, este roteiro de apresen-

tação foi realizado em três momentos distintos com um tempo médio de duração de noventa minutos para cada grupo.

5 Consolidar dados coletados:

Com o retorno de todos os questionários preenchidos, estes foram organizados e classificados a fim de viabilizar a consolidação dos dados fornecidos pelos participantes.

6 Análise preliminar dos resultados:

Esta etapa teve como base a avaliação dos dados consolidados e tratamento dos dados textuais a fim de realizar a análise de conteúdo e propiciar o cruzamento das percepções obtidas no Estudo de Campo com aquelas obtidas no Estudo de Caso.

6.4 Resultados do Estudo de Campo

Nesta seção são apresentados os resultados obtidos no Estudo de Campo. Estes resultados estão relacionados com o Objetivo 1 desta pesquisa. A apresentação destes resultados é realizada de acordo com a estrutura de questões utilizadas no formulários de coleta de dados.

Com este objetivo pretendeu-se identificar como os participantes percebem a influência da documentação nos seus processos de seleção de tarefas propostas na plataforma TopCoder. É importante enfatizar que a compreensão das tarefas que serão executadas pelos membros da multidão é vital para o desenvolvimento das tarefas. Corroborando com este pensamento Wieggers [40], que afirma que quando os requisitos representam o entendimento das necessidades dos envolvidos no projeto, e esta mesma compreensão é transferida para os desenvolvedores é ampliada a possibilidade de explorar melhor as possíveis alternativas de soluções para o requisito. Considerando que existem dois aspectos a serem avaliados a partir deste objetivo, estes serão analisados separadamente conforme segue:

Considerações sobre os fatores de seleção e descarte das tarefas

Com relação aos fatores que levam os participantes a selecionar ou descartar uma tarefa percebe-se que neste grupo a maior incidência de respostas diz respeito ao critério de “Clareza da Descrição”. Este fator se aplica tanto no processo de seleção utilizado pelos participantes, como também no de descarte de uma tarefa. A Tabela 6.3 demonstra quais foram os fatores mais relevantes para os participantes ao selecionar uma tarefa. Nesta mesma tabela pode-se observar que o fator com maior incidência para que um participante descarte uma tarefa também é a “Clareza da Descrição”. Deste modo, pode-se constatar que a compreensão da descrição de uma tarefa é um elemento crucial, uma vez que esta documentação é a efetiva representação do trabalho que este profissional deverá realizar.

A seguir são apresentados trechos de relatos efetuados pelos participantes que identificam os fatores que os levaram a selecionar uma determinada tarefa.

Tabela 6.3 – Critérios de Seleção/Descarte obtidos no Estudo de Campo

Critérios de Seleção das Tarefas		Critérios de Descarte das Tarefas	
Classificação	Nº de Ocorrências	Classificação	Nº de Ocorrências
Clareza na Descrição	5	Clareza na Descrição	4
Conhecer o Assunto	1	Desconhecer o Assunto	2
Desafio	1	Desconhecer a Tecnologia	1

Fonte: Elaborado pelo autor

- *“Clareza dos requisitos; Possibilidade de utilizar tecnologias recentes e “adquirir” experiência através da atividade; Esforço necessário em relação a recompensa” (P22)*
- *“Quando a tarefa possuía um nível de detalhamento técnico melhor. A questão de tarefas de ‘bugfix’ também chamou a atenção.” (P23)*
- *“Domínio da linguagem e familiaridade com o tipo de trabalho prestado em ambiente acadêmico e profissional.” (P25)*

Em relação aos critérios utilizados pelos participantes para descartar uma tarefa, pode-se destacar os seguintes relatos:

- *“Falta de detalhes importantes na documentação; Alta complexidade e requisitos não dão a clareza necessária; Impossibilidade de estimar o esforço em relação a recompensa.” (P22)*
- *“Tarefas muito abrangentes ou com documentação em muito alto nível.” (P23)*
- *“Falta de domínio da linguagem e consequentemente insegurança para os itens.” (P25)*

Observa-se que os fatores que envolvem a seleção de uma tarefa impactam de forma significativa no processo de execução da mesma, uma vez que muitas tarefas são analisadas pelos profissionais mas um número reduzido é realmente realizada e submetida à plataforma. Este fato pode ser constatado através dos dados exibidos na Tabela 6.4, em que é apresentada a listagem de todas as tarefas disponíveis para este Estudo de Campo com a indicação de quais foram avaliadas pelos participantes e quais foram selecionadas.

Para este estudo haviam 15 tarefa pré-selecionadas e sete participantes. Logo, com o produto destes dois elemento tem-se inicialmente 105 opções para a leitura das tarefas e mais 105 opções para a seleção das tarefas, conforme demonstrado na Tabela 6.4. Com estas informações pode-se constatar que:

- DA LEITURA: existe um total de 105 tarefas à disposição dos participantes para leitura (15 tarefas por participante). Deste conjunto, 58 tarefas foram efetivamente lidas, valor este que corresponde a 55% do total de tarefas disponíveis;

Tabela 6.4 – Seleção das tarefas pelos participantes do Estudo de Campo

ID	Tarefa	P22		P23		P24		P25		P26		P27		P28		TOTALS	
		L	S	L	S	L	S	L	S	L	S	L	S	L	S	L	S
TJ1	HTML Table data extraction - Ideation challenge	1	0	1	1	1	1	1	1	1	0	1	1	1	0	7	4
TJ2	Social Bike Owner App iOS - Backend Services Part 1	1	1	1	0			1	1	1	0	1	0	1	1	6	3
TJ3	XPrize Visioneers Native Android App Cache Implementation & Bug Fixes	1	1					1	0	1	0					3	1
TJ4	QED - Learning Library Mobile Application Ionic Prototype	1	0	1	1			1	0	1	1	1	0			5	2
TJ5	Panel Schedule Importer - Image De-Noise & In-Paint Machine Learning					1	0			1	0					2	0
TJ6	SM engine - CRUD Services Integration			1	1	1	1			1	1	1	1			4	4
TJ7	SM engine - Proxy Server Integration							1	1	1	1					2	2
TJ8	TC Review Microservice Local Deployment					1	0			1	1	1	1			3	2
TJ9	TC Identity Service - Groups API Return All Sub-groups					1	0			1	1	1	1			3	2
TA1	Holmes Negative News Web App Integration	1	0	1	0					1	0	1	1			4	1
TA2	[Payment Increased Again] Applet 2 HTML5 - Relationship Manager Bug Fixes	1	1	1	1	1	1			1	1	1	0			5	4
TA3	Predix - Integration with Alexa Voice Service	1	1	1	0	1	1			1	0					4	2
TA4	IBM Cognitive ChatBot - Misc Updates II	1	1	1	1					1	0			1	0	4	2
TA5	IBM Cognitive ChatBot - Misc Updates			1	1					1	0					2	1
TA6	Authentication Solution Angular4 and Material UI Prototype Challenge			1	1					1	1	1	0	1	1	4	3
Totais por Participante		8	5	10	7	7	4	5	3	15	7	9	5	4	2		

LEGENDA:

TJX: Identifica as tarefas cuja tecnologia predominante é Java.

TAX: Identifica as tarefas cuja tecnologia predominante é Angular.

L: Identifica que a tarefa foi lida pelo participante.

S: Identifica que a tarefa foi selecionada pelo participante para realizá-la.

Fonte: Elaborado pelo autor

- DA SELEÇÃO: deste total de 58 tarefas lidas, os participantes selecionaram 33 para efetivamente executá-las, o que corresponde a 30% do total de tarefas disponíveis;

A partir deste cenário, constata-se que das 15 tarefas disponíveis foram avaliadas em média 8,29 tarefas por participante, o que corresponde a aproximadamente 51% das tarefas disponíveis. Deste total de 15 tarefas, a média selecionada para execução cai para 4,71 tarefas, o que corresponde a aproximadamente 31% do total de tarefas disponíveis. Estes detalhes da análise podem ser visualizados na Tabela 6.5, que apresenta as informações consolidadas e os percentuais calculados.

Este fato relacionado ao número de possíveis submissões de tarefas que foi evidenciado pelo Estudo de Campo pode ser corroborado ao se observar as tarefas finalizadas na plataforma TopCoder, visto que em geral há um número elevado de inscrições em uma tarefa, mas um número reduzido de submissões, como pode ser observado na Figura 6.1. Esta figura apresenta um conjunto de seis fragmentos de tarefas finalizadas nos últimos meses na plataforma TopCoder. Nestes fragmentos que contém as descrições das tarefas estão destacadas em vermelho as inscrições e as submissões realizadas para elas. Neste conjunto de tarefas apresentadas há um total de 403 inscrições e apenas 44 submissões de soluções, o que indica que a média de submissões de soluções à plataforma nesta amostra é de aproximadamente 11%.

The figure displays seven screenshots of completed challenges on the TopCoder platform. Each screenshot shows the challenge title, tags, prize structure, status, and statistics. The statistics for 'REGISTRANTS' and 'SUBMISSIONS' are highlighted with red circles in each image.

Challenge Title	Registrants	Submissions
MongoDB Reporting UI	63	4
BG800 Online Course AngularJS Application Code Challenge	52	2
SRM 722 Editorials Contest - Div I - Easy - TCPPhoneHome	3	3
Ideation Mobile WebApp Design Challenge	66	9
Mjolnir Project - Ionic Hybrid App/ Node.js Backend Bug Bash	38	4
ConsenSys - uPort Blockchain and Identity Hackathon	177	22

Figura 6.1 – Fragmentos de tarefas finalizadas na plataforma TopCoder

Fonte: Elaborado pelo autor

Tabela 6.5 – Análise da seleção das tarefas pelos participantes no Estudo de Campo

Itens de Comparação	P22		P23		P24		P25		P26		P27		P28	
	L	S	L	S	L	S	L	S	L	S	L	S	L	S
Total de Tarefas identificadas por Participantes	8	5	10	7	7	4	5	3	15	7	9	5	4	2
Total de Tarefas JAVA identificadas por Participante	4	2	4	3	5	2	5	3	9	5	6	4	2	1
Total de Tarefas ANGULAR identificadas por Participante	4	3	6	4	2	2	0	0	6	2	3	1	2	1
% do Total Tarefas JAVA indentificadas por Participante	44%	22%	44%	33%	56%	22%	56%	33%	100%	56%	67%	44%	22%	11%
% do Total Tarefas ANGULAR indentificadas por Participante	67%	50%	100%	67%	33%	33%	0%	0%	100%	33%	50%	17%	33%	17%

LEGENDA:

L: Identifica que a tarefa foi lida pelo participante.

S: Identifica que a tarefa foi selecionada pelo participante para realizá-la.

Fonte: Elaborado pelo autor

Tabela 6.6 – Classificação da documentação pelos participantes do Estudo de Campo

Participante	Excelente	Muito boa	Boa	Fraca	Pobre
P22			x		
P23			x		
P24				x	
P25			x		
P26			x		
P27			x		
P28		x			

Fonte: Elaborado pelo autor

Considerações sobre a execução das tarefas

Como previsto, esta parte do objetivo da pesquisa não pôde ser observado neste Estudo de Campo, uma vez que o tempo disponível pelos participantes era reduzido e não permitiu a efetiva realização das tarefas pelos mesmos. Assim, neste estudo foi possível obter dados sobre os fatores que motivaram os participantes a selecionar as tarefas.

De modo geral como você classifica a documentação da(s) tarefa(s) que você selecionou para executar?

Para responder a esta questão foi solicitado que os participantes realizassem a avaliação de como cada um percebia a qualidade da documentação das tarefas fornecidas pela plataforma TopCoder. A avaliação deveria considerar o conjunto de todas as tarefas que os participantes tinham selecionado. Esta questão foi proposta aos participantes com o escopo fechado, ou seja, utilizando como opções de respostas a Escala Likert. O resultado obtido com esta questão pode ser visualizado na Tabela 6.6.

Baseado nos dados obtidos percebe-se que a maior parte dos participantes classificou a documentação como “Boa”. Isto indica que os participantes entenderam que as informações apresentadas eram suficientes para a execução das tarefas, exceto por dois participantes que manifestaram

opiniões divergentes, sendo eles: P24, que indicou que a documentação é “Fraca” e P28, que classificou a documentação das suas tarefas como “Muito boa”. Para estes dois casos que realizaram uma avaliação diferente do restante do grupo verificou-se que o participante P24 fez a leitura da documentação de 7 tarefas e selecionou 4 itens. Já o participante P28 fez a leitura da documentação de 4 tarefas e selecionou 2 itens.

Como você classifica a documentação da(s) tarefa(s) que você selecionou? Opine considerando cada um dos critérios a seguir e como ela se relaciona com a documentação fornecida.

Para responder a esta questão foi proposto um conjunto de critérios pré-definidos (Escala Likert) e assim cada um dos participantes deveria fazer a classificação de acordo com as tarefas que os mesmos selecionaram para executar. O resultado desta avaliação é apresentado na Tabela 6.7. Já o resultado tabulado das quantidades obtidas para a classificação de cada critério pode ser observado na Tabela 6.8. A seguir cada um destes critérios é analisado individualmente.

- **C1:** este critério indica se a documentação esta devidamente autocontida em sua descrição e documentos auxiliares. O resultado obtido é uma concentração de respostas apresentadas pelos participantes na opção “Concordo Parcialmente”.
- **C2:** trata do nível de detalhamento apresentado na descrição da tarefa. Neste caso, não houve um critério que se destacasse o resultado foi equilibrado entre quase todas as opções.
- **C3:** este critério trata do nível de precisão apresentado pela documentação das tarefas. Este critério apresentou uma distribuição do tipo normal para os critérios, o item de destaque é “Não Concordo Nem Discordo”.
- **C4:** este critério avalia se a documentação esta devidamente coesa, este item possui a concentração de respostas na opção “Concordo Parcialmente”.
- **C5:** indica se a documentação da tarefa propicia executar a tarefa dentro de um prazo exequível e dentro de um orçamento pré-definido. A maior incidência das respostas foram entre as opções “Concordo Totalmente” e “Concordo Parcialmente”.
- **C6:** revela se a documentação da tarefa não possui informações desnecessárias. Existe uma concentração das respostas nas opções “Concordo Totalmente” e “Concordo Parcialmente”.
- **C7:** questiona se a documentação proposta para a tarefa não apresenta informações dúbias. Houve uma concentração de respostas entre as opções “Concordo Parcialmente” e “Não Concordo Nem Discordo”.
- **C8:** indica se a documentação da tarefa proporciona dados suficientes para realizar os testes de validação necessários a estas tarefas. Neste critério houve uma distribuição de respostas semelhante entre as opções “Concordo Parcialmente”, “Não Concordo Nem Discordo” e “Discordo Parcialmente”.

Tabela 6.7 – Categorização da documentação pelos participantes do Estudo de Campo

Participante	C1	C2	C3	C4	C5	C6	C7	C8	C9
P22	CP	CT	NN	CP	CP	CP	CP	CP	CP
P23	CP	DP	NN	CP	DP	CT	CP	DP	CP
P24	CP	DP	DP	DP	DP	CP	CP	CP	CP
P25	CP	CP	NN	CP	CP	NN	NN	CP	CT
P26	CT	CT	CP	CP	NN	NN	CP	DP	NN
P27	CP	NN	NN	CP	DT	CT	NN	NN	NN
P28	CP	NN	CP	CP	NN	CT	NN	NN	NN

Legenda:

CT: Concordo totalmente
 CP: Concordo parcialmente
 NN: Não concordo nem discordo
 DP: Discordo parcialmente
 DT: Discordo totalmente
 NA: Não se aplica

Fonte: Elaborado pelo autor

Tabela 6.8 – Critérios de avaliação da documentação obtidos no Estudo de Campo

Critérios	Concordo totalmente (CT)	Concordo parcialmente (CP)	Não concordo nem discordo (NN)	Discordo parcialmente (DP)	Discordo totalmente (DT)	Não se aplica (NA)
C1	1	6	0	0	0	0
C2	2	1	2	2	0	0
C3	0	2	4	1	0	0
C4	0	6	0	1	0	0
C5	0	2	2	2	1	0
C6	3	2	2	0	0	0
C7	0	4	3	0	0	0
C8	0	3	2	2	0	0
C9	1	3	3	0	0	0
TOTAIS	7	29	18	8	1	0

Fonte: Elaborado pelo autor

- **C9:** trata da documentação e como esta poderá gerar problemas futuros. Houve praticamente uma divisão de respostas entre as opções “Concordo Parcialmente”, “Não Concordo Nem Discordo”.

Na sua opinião, quais pontos podem ser melhorados ou mantidos em relação a descrição da documentação (enunciado da tarefa e documentação complementar) associada à(s) tarefa(s) que você selecionou de forma a contribuir com a execução da mesma?

Para responder a essa questão foi solicitado aos participantes que após avaliarem as tarefas pré-selecionadas apresentassem pontos/aspectos em que a documentação pudesse ser melhorada considerando suas experiências e conhecimentos. Após realizada a análise dos dados obtidos, estes foram classificados. As sugestões de melhorias, bem como a classificação das informações estão apresentadas na Tabela 6.9. Constatou-se que cinco dos sete participantes indicam que a documentação da tarefa deve ser mais clara e detalhada. Algumas das sugestões são explícitas, como a

apresentada pelo participantes P22. Em outros casos, diferentes fatores envolvidos na documentação acabam por refletir em uma documentação mais clara e objetiva, estes casos foram relatados pelos participantes a seguir indicados: (i) P24, P25 e P26, que sugerem a utilização de diagramas ou *screenshots* como forma de complementar as informações da documentação, (ii) P27, que sugere que as estruturas complementares que envolvem a configuração do ambiente fiquem explicitadas na documentação de modo a simplificar a sua utilização no desenvolvimento e (iii) P28, que sugere a apresentação de itens utilizados na plataforma com a indicação de ferramentas e a utilização de padrões de código para auxiliar no desenvolvimento. Além da clareza da documentação há a sugestão de padronizar as tarefas com o propósito de se ter uma unidade de informações apresentadas nas tarefas. Esta sugestão foi apresentada de forma explícita pelo participante P23 e também mencionada pelo participante P24.

6.5 Discussão sobre o Estudo de Campo

Com o desenvolvimento deste Estudo de Campo foi possível constatar que o grupo de participantes possui percepções semelhantes quando se trata de determinados aspectos relatados através do Estudo Caso. Cabe reforçar que neste Estudo de Campo os participantes foram pré-selecionados e possuíam perfis similares (conhecimentos e experiência prévias na área de TI e em determinadas linguagens de programação), o que pode fazer com que eles tenham um mesmo tipo de entendimento sobre as questões apresentadas.

No que diz respeito às características intrínsecas ao Estudo de Campo apresentadas por McGrath [29] anteriormente, destaca-se a relação existente entre elas, de tal forma que ao se potencializar uma das características, outras serão atenuadas. Logo, nesta investigação pode-se perceber que: (i) a Generalização advinda dos participantes provê uma diversidade de opiniões que surge da experiência individualizada dos mesmos, essa diversidade reduz esta característica em consequência do número de participantes, (ii) a Precisão em relação à medição do fenômeno foi relativamente controlada, uma vez que o conjunto de tarefas propostas aos participantes foi restrita, fato que não ocorreu no Estudo de Caso, e (iii) o Realismo esperado para esta coleta de dados foi atendido, uma vez que as tarefas propostas aos participantes foram extraídas da plataforma TopCoder e os participantes possuíam o conhecimento necessário para a execução das mesmas. A partir deste cenário tem-se a característica do Realismo com maior ênfase, seguida da Precisão e por último e mais atenuada está a Generalização.

Com isso, observa-se que quando se trata da capacidade de avaliação das tarefas, principalmente quando estas tarefas estão diretamente relacionadas aos conhecimentos dos participantes, a experiência tem uma influência direta no entendimento da tarefa e conseqüentemente na capacidade de seleção e execução da mesma. Talvez por ser um grupo de profissionais com experiência no desenvolvimento de sistemas muitas observações reforçaram a necessidade do entendimento da tarefa e da necessidade de uma documentação correta e clara. Estes são fatores que propiciam o desenvolvimento das tarefas de modo ágil e eficiente.

Tabela 6.9 – Sugestões de melhorias para a documentação das tarefas - Estudo de Campo

Participante	Declaração	Mapeamento Modelo
P22	<ul style="list-style-type: none"> • Mais clareza e objetividade nos requisitos de negocio • Melhoria na descrição dos requisitos não funcionais. 	<ul style="list-style-type: none"> • Detalhamento da Tarefa
P23	<p>Acho que existe uma grande variedade de formato de descrição das tarefas. Poderiam tentar criar um padrão no site.</p> <p>Algumas tarefas vão contra a ideia de pequenas tarefas, algumas como TJ2 e TA3 poderiam ser quebradas em tarefas menores. Além disso, tarefas como a TA5 e TA4 são tarefas menores e melhores descritas e ainda possuíam um prazo mais realista de entrega.</p>	<ul style="list-style-type: none"> • Padronização • Detalhamento da Tarefas
P24	Acho que poderia ter mais diagramas para auxiliar no entendimento , prints de telas ou wireframes e também a própria plataforma TopCoder exigisse uma mínima padronização da documentação .	<ul style="list-style-type: none"> • Padronização • Detalhamento da Tarefa • Documentação Complementar
P25	Diagramas e descrição estruturada dos passos. Ilustrar ou adotar um método mais lúdico pode economizar tempo na interpretação da tarefa.	<ul style="list-style-type: none"> • Detalhamento da Tarefa • Documentação Complementar
P26	<p>No documento TJ4, os screenchots ajudam bastante a ter uma visão ampla do trabalho.</p> <p>No documento TJ8 poderiam sinalizar onde fisicamente esta o readme, pode estar na raiz (padrão) mas também pode não estar.</p> <p>No documento TA2 possui o passo-a-passo de reprodução isso ajuda bastante.</p> <p>No documento TA6 os links complementares auxiliam no desenvolvimento e deixam a tarefa completa na descrição.</p>	<ul style="list-style-type: none"> • Configuração do Ambiente • Links Relacionados • Detalhamento da Tarefa • Documentação Complementar
P27	<p>Algumas tarefas deveriam liberar um esboço do projeto (arquivos compactados) contendo a estrutura de pastas a fim de orientar o desenvolvimento (TA1)</p> <p>Os prazos das tarefas deveriam ser maiores, e se for o caso, dar bônus por entregas antecipadas.</p>	<ul style="list-style-type: none"> • Configuração do Ambiente • Documentação Complementar
P28	Acho que a definição das ferramentas a serem utilizadas e os padrões de código são pontos fundamentais e devem ser mantidos.	<ul style="list-style-type: none"> • Padronização • Ferramentas

Fonte: Elaborado pelo autor

6.6 Limitações do Estudo de Campo

Ao realizar este Estudo de Campo foram identificados alguns aspectos que merecem atenção e que devido às condições e prazos para a execução da pesquisa foram limitantes, são eles:

- Disponibilidade de tempo dos profissionais para realizar a tarefa de forma completa, ou seja, selecionar e executar a tarefa até a submissão da mesma na plataforma. Tendo em vista que praticamente todas as tarefas demandam um esforço significativo, obter o engajamento de uma equipe de profissionais para realizar todo o processo é difícil sem um aporte financeiro;

- O número de participantes do Estudo de Campo poderia ser maior para que se pudesse obter uma amostragem ainda mais significativa e precisa das opiniões destes profissionais;
- O Estudo de Campo foi iniciado muito próximo ao o Ciclo 2 do Estudo de Caso. Desta forma o Estudo de Campo foi praticamente executado em paralelo. Devido aos prazos reduzido para finalização da pesquisa não foi possível analisar os resultados obtidos no Estudo de Caso (ciclo 1 e 2), para então iniciar o Estudo de Campo.
- O conjunto de conhecimentos/experiências dos participantes poderia ser ampliado, pois os conhecimentos dos profissionais ficaram limitados às linguagens Java e Angular e suas experiências na área de TI eram semelhantes. Idealmente seria oportuno obter um conjunto mais amplo de conhecimentos/experiências profissionais.

7. MODELO PARA DOCUMENTAÇÃO DE TAREFAS NA TOPCODER

Neste capítulo é apresentada um modelo de documentação da tarefa para a plataforma TopCoder. Esta proposta foi construída com base nos resultados obtidos no Estudo de Caso e no Estudo de Campo e também no material científico relativo ao tema de pesquisa. Este modelo esta diretamente relacionado ao Objetivo 2 desta pesquisa.

7.1 Contexto

Em diversos momentos durante a realização desta pesquisa foram relatados problemas envolvendo a documentação da tarefa, sejam eles decorrentes de uma construção de texto pobre ou confusa, de conteúdos abordados pela documentação que não são compreendidos, ausência de informações dentre outros problemas. Este fato é confirmando por Machado e colegas [25], quando cita em seu trabalho a existência de dificuldades e problemas que envolvem o *Software Crowdsourcing*, dentre estes, salienta-se o problema com a documentação das tarefas que são fornecidas aos membros da multidão.

Como mencionado na Seção 2.3 do Capítulo 2, a documentação fornecida para a tarefa é decorrente da fragmentação dela. Quando o contratante propõe uma tarefa à plataforma, esta será avaliada e conforme a necessidade deverá ser fragmentada em micro tarefas para que seja executada de forma mais eficiente. Do mesmo modo como há uma avaliação da tarefa, deve ocorrer uma avaliação da documentação fornecida pelo contratante para que este material seja adequada às suas micro tarefas conforme mencionado no Capítulo 2 desta pesquisa.

Um problema envolvendo a não decomposição da documentação da tarefa foi observada pelo pesquisador quando da construção do instrumento aplicado no Estudo de Campo. Ao realizar a seleção de tarefas que seriam utilizadas no estudo verificou-se que duas tarefas que estavam correlacionadas apresentavam exatamente a mesma documentação. Em certa medida as informações adicionais podem auxiliar no desenvolvimento da tarefa, mas por outro lado estas informações também podem causar dúvidas ou até transtornos para a sua execução. As tarefas que possuem a mesma documentação são "*SM engine - CRUD Services Integration*" e "*SM engine - Proxy Server Integration*". Estas duas tarefas dispõem do mesmo documento de especificação funcional, e este, por sua vez, possui uma abrangência maior do que a apresentada por cada uma das tarefas.

Com relação ao conjunto de informações fornecidas pelas tarefas, pode-se dizer que existe uma grande variabilidade em sua estrutura e principalmente em seu conteúdo. Esta variação pode ser ilustrada através da Figura 7.1, que apresenta seis tarefas distintas com significativas diferenças em suas descrições. Como pode ser observado, a quantidade de informações apresentadas na Tarefa A é muito superior e apresentada na Tarefa F. Nesse sentido, é compreensível que nem todas as tarefas possam ter o mesmo volume de dados, uma vez que tratam de problemas diferentes, mas o

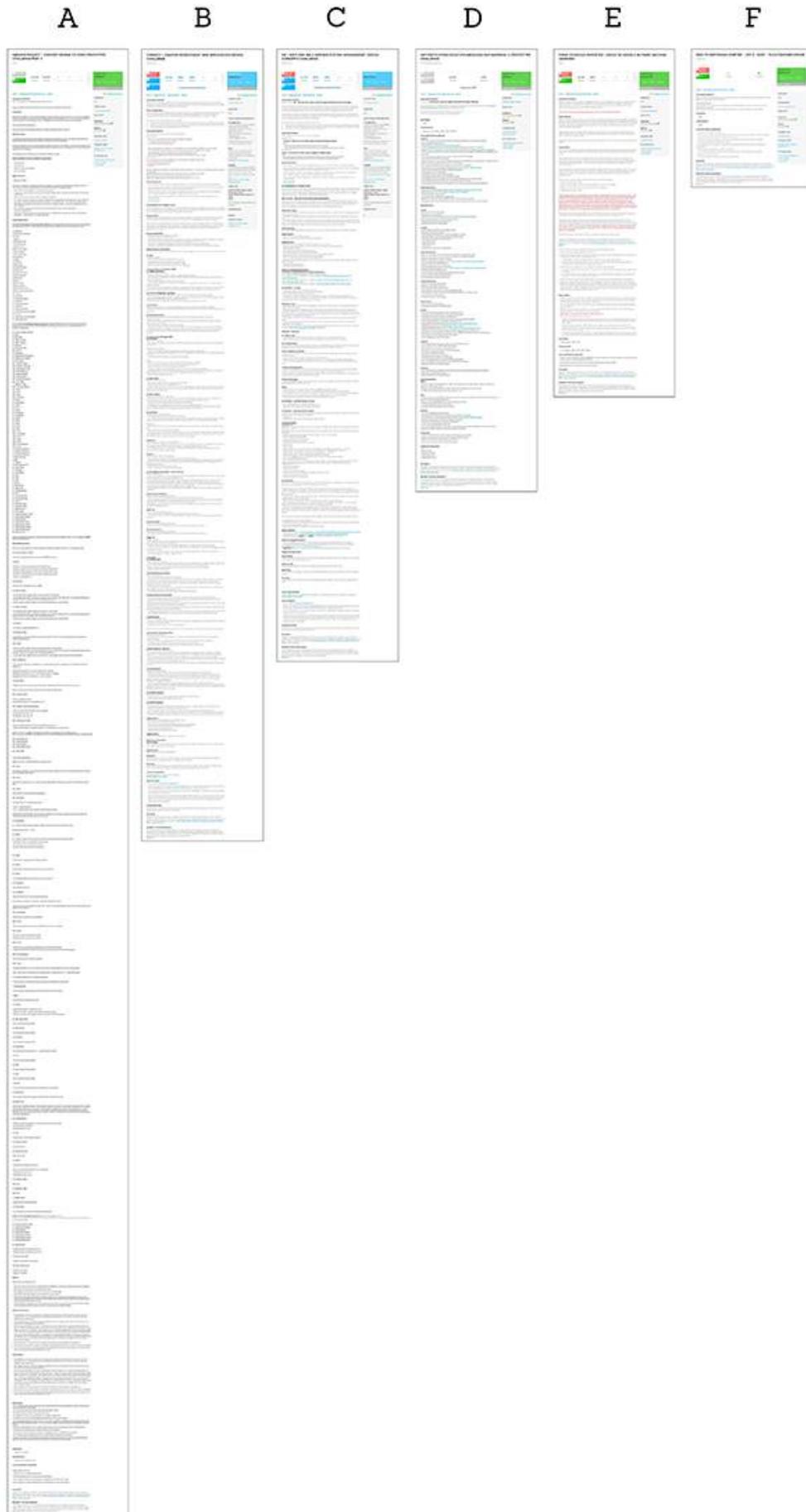


Figura 7.1 – Comparativo da descrição de tarefas na plataforma TopCoder

Fonte: Elaborado pelo autor

volume de informações deve ser suficiente e razoável para a sua execução. Os dados de identificação de cada uma das tarefas apresentadas na Figura 7.1 são listados a seguir:

- A *MJOLNIR Project - Convert Design to Ionic Prototype Challenge Part 2* [CODE]
- B *SYNERGY - Campus Recruitment Web Application Design Challenge* [Web Design]
- C *GE - Next Gen Air & Surface System Management Design Concepts Challenge* [Web Design]
- D *Authentication Solution Angular 4 and Material UI Prototype Challenge* [UI Prototype Competition]
- E *Panel Schedule Importer - Image De-Noise & In-Paint Machine Learning* [CODE]
- F *SRM 721 Editorials Contest - DIV II - Easy - FlightDataRecorder* [First2Finish]

7.2 Modelo de Documentação

Nesta seção é apresentada uma proposta para a documentação das tarefas na plataforma TopCoder. Para a construção desta proposta foram analisados os resultados obtidos através do Estudo de Caso e do Estudo de Campo. Além do material extraído dos dois estudos foram utilizados materiais científicos relacionados ao tema de pesquisa, dentre eles pode-se mencionar o trabalho de Machado [25], que trata sobre a temática de documentação das tarefas.

De modo geral, a estrutura da documentação das tarefas proposta pela plataforma TopCoder apresenta uma padronização que é relativamente aberta, de forma a tornar mais simples a sua gestão e publicação. Esta estrutura de documentação varia de acordo com o tipo de desafio, mas em linhas gerais apresenta elementos básicos, conforme a lista apresentada a seguir:

- **Cabeçalho da Tarefa:** apresenta diversas informações relacionadas à tarefa, tais como: Título, tecnologias, premiação (valores e número de premiações), prazos e, por fim, determinados links (detalhes da tarefa, inscrições, acesso ou fórum, dentre outros).
- **Challenge Overview ou Summary:** neste item da documentação é exibido um resumo da tarefa que pode variar de acordo com o tipo de tarefa. Em diferentes casos este elemento já apresenta a descrição das atividades da tarefa;
- **Payments:** neste ponto da documentação são expostas as regras relacionadas ao pagamento dos prêmios vinculados à tarefa;
- **Reliability Rating and Bonus:** apresenta um texto genérico que informa como é realizada a contagem de pontos para os participantes.

Existem algumas seções da documentação da tarefa que aparentemente são variáveis conforme o contratante ou talvez por segmentação, uma vez que nem sempre estão disponíveis nas tarefas da plataforma. A seguir são apresentadas alguns desses casos:

- **Plataforms:** nesta seção são indicadas as plataformas utilizadas na realização da tarefa, porém nem sempre estão disponíveis na declaração da tarefa. Por exemplo: Microsoft Azure, Linux;
- **Technologies:** nesta seção são descritas as tecnologias utilizadas na realização da tarefa, entretanto não são sempre apresentadas. Por exemplo: C#, Docker, Java;
- **Final Submission Guidelines:** nesta seção são apresentadas instruções relacionadas à submissão da tarefa. Este tipo de orientação nem sempre está disponível nas tarefas.

Considerando a estrutura de tarefa atualmente apresentada pela plataforma TopCoder e os resultados obtidos através dos dois estudos, foi construído um modelo para a documentação da tarefa que visa aprimorar a estrutura da descrição das tarefas conforme apresentado na Figura 7.2. Assim, a partir dos relatos dos participantes buscou-se elementos que pudessem contribuir na melhoria desta documentação, são eles: (i) qualidade da informação apresentada na documentação da tarefa, seja na própria descrição da tarefa ou mesmo em documentação complementar, e (ii) padronização da tarefa, identificação de ferramentas e informações complementares, dentre outras.

É importante reforçar que este trabalho propõe um modelo de documentação para as tarefas, ou seja, apresenta uma estrutura lógica para as informações com o objetivo de padronizar o documento e facilitar a sua leitura e entendimento. Entretanto, esta pesquisa não define a forma como a informação será descrita na tarefa, isto é, não determina um padrão de escrita para a documentação. Assim, é fundamental que a documentação siga um conjunto de regras e padrões na sua forma de registrar os fatos (textualmente ou com o uso de elementos gráficos), de modo a tornar a informação clara e objetiva a todos que se propuserem a ler tal documentação.

Não foi proposto nenhum tipo de alteração para o cabeçalho de identificação da tarefa nem para a área de informações. Os ajustes propostos dizem respeito à estrutura do corpo da descrição da tarefa. Este modelo de documentação da tarefa está diretamente alinhado com os relatos dos participantes, como pode ser observado na Tabela 7.1. Esta padronização foi realizada através dos tópicos destacados no corpo do documento conforme apresentado na Figura 7.2. Cada elemento do documento foi identificado por um ícone específico. Também foi utilizado um padrão de cores para representar os itens já existentes, os alternativos e os novos elementos. Esta identificação foi realizada da seguinte forma: (i) Preto, são os tópicos “padrão” da documentação atual utilizada na plataforma e devem ser mantidos, (ii) Azul, os ícones representam elementos que são apresentados em determinados documentos, mais frequentemente em UI ou Web e (iii) Verde, são os elementos sugeridos, que compõem a modelo de documentação deste trabalho. Cada um destes elementos será descrito a seguir:

Título da Tarefa

Premiação

1st	2nd	3rd	4th	5th
\$ 1500	\$ 1000	\$ 700	\$ 400	\$ 100

Próximo Marco Registro | 10d 12h

Registrar Submeter

Exibir Detalhes

Detalhes Registros (10)

Resumo da Tarefa

Plataformas

Tecnologias

Links Relacionados

Configuração do Ambiente

Pré Requisitos

Tarefas Relacionadas

Ferramentas

Detalhamento da Tarefa

Documentação Complementar

Critérios de Teste e Aceitação

Regras de Submissão da Tarefa

Pagamentos

Ranking Confiabilidade e Bônus

Informações

Supporte

Cabeçalho

Corpo

Figura 7.2 – Proposta de documentação de uma tarefa para a plataforma TopCoder

Fonte: Elaborado pelo autor



Resumo da Tarefa

Este elemento já é utilizado na plataforma. Sua nomenclatura pode variar conforme o tipo de tarefa.

Tabela 7.1 – Classificação dos elementos utilizados no modelo de documentação

Classes do Modelo	Participante
Padronização	P7, P14, P18, P19, P20, P23, P24, P28
Resumo da Tarefa	P3, P9
Plataformas	
Tecnologias	P3
Links Relacionados	P3, P7, P8, P11, P16, P26
Configuração do Ambiente	P7, P10, P11, P13, P15, P18, P20, P26, P27
Pré-Requisitos	P2, P8
Tarefas Relacionadas	P3, P7
Ferramentas	P8, P11, P17, P20, P21, P28
Detalhamento da Tarefa	P1, P3, P4, P5, P6, P8, P9, P10, P11, P12, P13, P15, P16, P17, P18, P20, P22, P23, P24, P25, P26
Documentação Complementar	P1, P7, P8, P11, P14, P16, P17, P24, P25, P26, P27
CrITÉrios de Teste e Aceitação	P3, P4, P7, P9, P10, P12, P13, P17, P18, P20
Regras de Submissão da Tarefa	P7, P9, P10, P15, P17

Fonte: Elaborado pelo autor

Com maior frequência é denominado “*Challenge Overview*”, também podendo ser identificado como “*Challenge Summary*”.



Plataformas

Este elemento já é utilizado na plataforma em determinadas tarefas. A proposta é que este item passe a ser padrão em todas as tarefas que apresentem a necessidade de desenvolvimento de software ou que o resultado da tarefa seja aplicável a um ambiente específico. Sua nomenclatura na plataforma é “*Platforms*”.



Tecnologias

Do mesmo modo que o elemento anterior, a indicação das tecnologias é apresentada em algumas tarefas e em outras não. Nesta proposta passa a ser um item padrão em todas as tarefas, indicando quais tecnologias estão envolvidas na execução da tarefa. Sua nomenclatura na plataforma é “*Technologies*”.



Links Relacionados

Este item foi proposto em função dos participantes relatarem que determinadas tarefas possuíam no texto da descrição ou mesmo nos documentos complementares a indicações de *links* externos à plataforma. Esses *links* tinham como objetivo apresentar esclarecimentos diversos, como por exemplo, fornecer acesso a tutoriais ou aplicações que estavam relacionadas com a resolução da tarefa. Esta sugestão visa agrupar em um único local as informações adicionais de modo a simplificar sua identificação, assim como seu acesso por parte dos membros da multidão que estiverem acessando esta tarefa.



Configuração do Ambiente

Em geral, para que seja possível executar a maior parte das tarefas de desenvolvimento é necessário que os membros da multidão preparem o ambiente para a realização da tarefa. Em determinados casos a montagem segue o padrão das ferramentas ou *frameworks*, entretanto, em outros casos a preparação deste ambiente requer a utilização de componentes especiais que demandam um tempo maior e um cuidado ao executá-lo. Com base neste cenário é sugerido este elemento, onde devem ser apresentados os requisitos de configuração do ambiente com a indicação dos *links* necessários para a sua construção. Estas informações auxiliam inclusive os membros da multidão a compreender o esforço necessário para a sua montagem, como também o hardware necessário para tal.



Pré-Requisitos

Este elemento apareceu principalmente nos relatos dos participantes que a executaram uma tarefa envolvendo o uso de *Bitcoins Ethereum*. Esta tarefa necessitava de uma *Wallet Ethereum*, que não foi mencionada na documentação e que impactou na execução da mesma. Este é um item que surge entre os relatos finais e foi proposto pelos participantes P2 e P8. Acredita-se que este tipo de problema não seja apenas um caso isolado e que pode ocorrer em outras tarefas. Desta forma, este item pode tornar-se padrão para a documentação com o intuito de torná-la mais precisa e completa.



Tarefas Relacionadas

Este elemento possui sua origem em conversas com os participantes da pesquisa e a sugestão dos participantes P3 e P7. Nestes relatos, os participantes mencionaram que em muitas tarefas tenta-

ram buscar outras tarefas que talvez estivessem associadas a que eles estavam avaliando, de modo a permitir uma compreensão mais adequada das necessidades do projeto. Entretanto, o mecanismo de da plataforma TopCoder torna essa ação bastante restrita, uma vez que é necessário ter conhecimento do nome das tarefas, o que nem sempre pode ocorrer. Neste sentido, foi proposta a inclusão deste elemento que estaria disponível em todas as tarefas que possuísem tarefas correlacionadas, simplificando assim o processo de pesquisa e o conhecimento das necessidades propostas.



Ferramentas

Esse item foi proposto para que todas as ferramentas necessárias para a realização da tarefa fossem identificadas, assim como os *links* de acesso às suas documentações ou download das mesmas. Este elemento foi comentado em diferentes momentos por diversos participantes, mas registrado como sugestão por P8, P11, P17, P20, P21 e P28.



Detalhamento da Tarefa

Este é o elemento essencial para a realização da tarefa, afinal é ele que efetivamente irá apresentar aos membros da multidão o que deverá ser realizado, isto é, representa os requisitos da tarefa a ser executada. Logo, a sua clara redação é o principal fator de sucesso para a realização da tarefa. Sendo assim, acredita-se que este item deve possuir as características indicadas na Seção 2.4 do Capítulo 2, que descreve as características dos requisitos, apresenta critérios de qualidade e um guia de boas práticas para a construção de requisitos com qualidade.

O maior volume de comentários apresentados pelos participantes faz referência à qualidade do texto registrado na tarefa, o que em muitos casos é confuso, pouco claro, evasivo ou não apresenta as informações necessárias. Estes relatos foram registrados pelos participantes P1, P3, P4, P5, P6, P8, P9, P10, P11, P12, P13, P15, P16, P17, P18, P20, P22, P23, P24, P25 e P26. Desse modo, reforça-se que o ponto crucial da documentação volta-se à pessoa que esta redigindo ou ajustando a documentação para publicação na plataforma, que deve realizar um trabalho focado para manter a alta qualidade das informações prestadas. Acredita-se que este cuidado com a documentação poderá trazer resultados positivos no volume de submissões para as tarefas, o que tende a trazer também uma melhor qualidade nas entregas realizadas pelos participantes. Assim, cabe mencionar que existem tarefas bem documentadas e estas foram identificadas pelos participantes desta pesquisa, conforme demonstrado nos relatos realizados pelos participantes P2, P6, P8, P9, P18 e P26.



Documentação Complementar

Atualmente os documentos complementares das tarefas ficam disponíveis no fórum ou em *links* dentro da descrição da documentação. A proposta é tornar este um elemento padrão da documentação para que todas as informações necessárias à realização das tarefas (especificações, diagramas, modelos, imagens, áudios, vídeos, entre outros itens) fiquem centralizadas em um único local. Esta prática visa tornar mais rápida a identificação dos documentos auxiliares, bem como permite aos participantes identificar o volume de informações que estão envolvidas na execução da tarefa. Outro aspecto importante é que sendo um elemento padrão, não se torna mais obrigatório aos participantes realizar a subscrição na tarefa para poder acessar o fórum e só então ter acesso a estas informações. Este item representa os relatos realizados pelos participantes P1, P7, P8, P11, P14, P16, P17, P24, P25, P26 e P27.



Critérios de Teste e Aceitação

Como um fator principal é a identificação de quais são os critérios de aceitação da tarefa, é sugerido que sejam identificados explicitamente quais são os eventuais testes ou roteiros de testes esperados para a realização da tarefa. Com estas informações os membros da multidão podem saber antes mesmo de iniciar a tarefa o que é esperado para a sua entrega de forma objetiva. Assim como outros itens propostos, a ideia deste elemento surgiu inicialmente a partir de comentários realizados durante a execução do Estudo de Caso. Alguns dos participantes que registraram este tipo de sugestão foram: P2, P3, P4, P7, P9, P10, P12, P13, P17, P18 e P20.



Regras de Submissão da Tarefas

Este elemento é apresentado em algumas tarefas. Considerando-se que ele deve descrever o que deve ser submetido e como deve ocorrer esta submissão da solução, avalia-se que este deve ser um elemento obrigatório em todas as tarefas. Tendo as regras claras tornam-se mais objetivas e eficientes as atividades dos membros da multidão. Os participantes que sugeriram este tipo de registro na documentação foram: P2, P7, P9, P10, P15 e P17.



Pagamentos

Este elemento da documentação consta no padrão da documentação, deve ser mantido inalterado.

O nome deste item na plataforma é “*Payments*”.



Ranking de Confiabilidade e Bônus

Representa um dos elementos já existentes na documentação padrão. Deve ser mantido sem alterações. O nome deste item na plataforma é “*Reliability Rating and Bonus*”.

Concomitante aos ajustes propostos na documentação de tarefas, existem ainda algumas sugestões feitas pelos participantes da pesquisa que podem ser incorporados a todos os elementos da documentação, são elas:

- Utilizar mídias audiovisuais como apoio à documentação, estas mídias podem ser disponibilizadas através de *links* exibidos no texto de qualquer um dos elementos propostos. Esta sugestão foi feita pelos participantes P7 e P11.
- Acrescentar a documentação de arquitetura da aplicação a fim de auxiliar no desenvolvimento da tarefa, este item poderia estar descrito na “Documentação Complementar” ou mesmo no “Detalhamento da Tarefa”. Este item foi sugerido pelos participantes P12 e P17.
- Construir a documentação da tarefa com base em um *checklist*, a padronização proposta traz um pouco desta ideia de *checklist*, mas talvez ainda assim possa ser desenvolvido algum tipo de *checklist* que dê suporte à construção do detalhamento da tarefa em si. Esta sugestão foi feita pelo participante P19.

Outras sugestões que foram indicadas pelos participantes fazem referência à própria plataforma TopCoder e sua operação, sendo elas:

- Disponibilizar a impressão em formato PDF do texto da documentação de forma completa, assim o membro da multidão não precisa estar *on-line* para realizar a leitura da mesma. Esta funcionalidade parece ser bem interessante considerando que em vários momentos durante a realização desta pesquisa a plataforma não estava respondendo aos acessos. Esta sugestão foi feita pelo participante P3.
- Bonificar os membros da multidão que submeterem seu resultado antecipadamente e que sua resposta seja uma finalista ou a vencedora. Tendo em vista que os prazos para a execução das tarefas em geral são curtos (conforme relatos recorrente principalmente dos participantes do Estudo de Caso), esta ação poderá inclusive aumentar a eficiência nas entregas e proporcionar mais motivação para os membros da multidão realizarem tais tarefas. Esse item foi sugerido pelo participante P27.

Tabela 7.2 – Critérios de classificação de requisitos aplicados ao modelo de documentação

ID	Critérios de Classificação de Requisitos	Elementos do Modelo												
		Padronização	Resumo da Tarefa	Plataformas	Tecnologias	Links Relacionados	Configuração do Ambiente	Pré-Requisitos	Tarefas Relacionadas	Ferramentas	Detalhamento da Tarefa	Documentação Complementar	Critérios de Teste e Aceitação	Regras de Submissão da Tarefa
C1	A documentação fornecida para a sua tarefa estava devidamente autocontida , ou seja, não necessita de outras informações para executar a tarefa.	●	●	●	●	●	●	●	●	●	●	●	●	●
C2	A documentação fornecida para a sua tarefa possui o nível de detalhamento necessário para executar a tarefa, ou seja, descreve as etapas que devem ser realizadas de forma satisfatória para a finalização da tarefa.	●	●				●	●		●	●	●	●	●
C3	A documentação fornecida para a sua tarefa possui o nível de precisão necessária para executar a tarefa, ou seja, descreve os detalhes de como os procedimentos devem ser realizados para finalizar a tarefa.	●	●				●	●		●	●	●	●	●
C4	A documentação fornecida para a sua tarefa está corretamente documentada e coesa de forma a dar o suporte necessário para executar a tarefa.	●	●								●	●		
C5	A documentação fornecida para a sua tarefa descreve uma tarefa factível dentro do prazo e orçamento proposto.	●	●								●			
C6	A documentação fornecida para a sua tarefa não possui informações desnecessárias para a execução da tarefa.	●	●	●	●	●	●	●	●	●	●	●	●	●
C7	A documentação fornecida para a sua tarefa possui informações dúbias , ou seja, pode apresentar interpretações divergentes conforme o interlocutor que estiver lendo esta documentação.	●				●	●	●	●	●	●	●	●	●
C8	A documentação fornecida para a sua tarefa possui informações suficientes para realizar os testes necessários a fim de validar se a tarefa foi cumprida conforme esperado.	●											●	●
C9	A documentação fornecida para a sua tarefa possui problemas que podem gerar defeitos no resultado final da tarefa.	●	●								●	●	●	●

Fonte: Elaborado pelo autor

Como forma de verificar se os elementos indicados no modelo apresentavam uma cobertura em relação aos critérios de qualificação da documentação utilizados no Estudo de Caso e no Estudo de Campo, foi realizado um mapeamento destes critérios e da forma como cada um dos elementos está associado a eles. A partir deste mapeamento observou-se que o modelo possui aderência a estes critérios.

Conforme já apresentado anteriormente, baseado na análise dos dados obtidos no Estudo de Caso, no Estudo de Campo e na Revisão da literatura, foi possível propor um modelo de documentação que viesse a atender as necessidades de informações que os membros da multidão precisam para realizar uma tarefa. Este modelo visa apresentar a estrutura de documentação da tarefa, tendo como premissa a padronização destas informações a fim de tornar mais lógica e eficiente a identificação das mesmas.

Cabe destacar que o elemento do modelo relativo à Padronização foi vinculado a todos os critérios. Esta associação ocorreu devido ao seu caráter de adequação das informações da tarefa, ou seja, esse elemento pretende contribuir para a melhoria da compreensão das informações apresentadas

na tarefa. A distribuição das informações em segmentos específicos no modelo proposto permitirá aos membros da multidão compreender de forma mais rápida e efetiva o objetivo da tarefa.

A forma como cada elemento do modelo se propõe a atender aos critérios de avaliação podem ser observados na Tabela 7.2. Este mapeamento é detalhado a seguir:

- C1 e C6

Estes critérios destacam características relacionadas a completude da documentação (C1) e a necessidade de não possuir informações desnecessárias (C6). Neste sentido, entende-se que o fracionamento dos dados entre todos os elementos apresentados no modelo contribui para o alcance destas características, uma vez que, se estas informações estão devidamente contidas em cada elemento do modelo, a informação tenderá a ser mais precisa e completa em cada elemento. Além disso, cada elemento, por possuir um conteúdo mais restrito, permite uma validação mais rápida e eficiente.

- C2 e C3

Por possuírem características muito próximas e correlatas no seu entendimento, os critérios de nível de detalhamento (C2) e nível de precisão (C3) foram agregados sob os mesmos elementos do modelo (Resumo da Tarefa, Configuração do Ambiente, Pré-Requisitos, Ferramentas, Detalhamento da Tarefa, Documentação Complementar, Critérios de Teste e Aceitação e Regras de Submissão da Tarefa). Estes elementos do modelo descrevem as tarefas sob diversos aspectos, desde critérios técnicos para a sua execução, como por exemplo a Configuração do Ambiente ou Ferramentas, ou mesmo questões que envolvem a própria descrição do que deve ser realizado, por exemplo, Detalhamento da Tarefa ou Critérios de Teste e Aceitação e Regras.

- C4

Este critério tem como finalidade descrever se a documentação está coesa em sua estrutura e informações prestadas. Neste sentido, os elementos de Resumo da Tarefa, Detalhamento da Tarefa e Documentação Complementar devem encapsular tal conhecimento a fim de descrever explicitamente os objetivos a serem realizados pela tarefa.

- C5

Em relação a este critério, que indica se um requisito é factível, tem-se como elementos relacionados o Resumo da Tarefa e o Detalhamento da Tarefa. Estes elementos descrevem o que deverá ser realizado na tarefa. Desta forma o membro da multidão poderá, de forma direta, identificar se a tarefa pode ou não ser realizada conforme sua descrição.

- C7

Este critério, que trata da apresentação de informações dúbias, foi associado aos seguintes elementos do modelo: Links Relacionados, Configuração do Ambiente, Pré-Requisitos, Tarefas Relacionadas, Ferramentas, Detalhamento da Tarefa, Documentação Complementar, Critérios de Teste e Aceitação e Regras de Submissão da Tarefa. Estes elementos do modelo pretendem

concentrar determinadas informações a fim de que elas não se tornem repetitivas em outros elementos. Por exemplo, uma vez descrito o procedimento para a configuração do ambiente não é mais necessário realizar tal descrição no detalhamento da tarefa. Esta mesma lógica foi aplicada aos demais elementos aqui propostos.

- C8

Com relação ao critério relativo às informações suficientes para realizar os testes envolvidos na tarefa, o modelo possui os elementos de Critérios de Teste e Aceitação e Regras de Submissão da Tarefa, os quais destinam-se a suprir especificamente este tipo de lacuna. Isso porque o primeiro elemento concentra as informações específicas para a validação do que foi desenvolvido como solução para a tarefa e o segundo registra como esta tarefa deverá ser submetida à plataforma.

- C9

O critério referente à identificação de problemas envolvendo a documentação de requisitos possui uma relação direta com os seguintes elementos: Resumo da Tarefa, Detalhamento da Tarefa, Critérios de Teste e Aceitação e Regras de Submissão da Tarefa. Este conjunto de elementos pode ser subdividido em dois grupos: o primeiro grupo descreve objetivamente o que deverá ser realizado pela tarefa e o segundo grupo trata da forma como deve ser validada a solução da tarefa e sua submissão à plataforma. Isso significa que ao realizar a análise dos elementos contidos no primeiro grupo, pode-se perceber inconsistências intrínsecas ao seu detalhamento, já no segundo grupo existem as regras de validação que podem ser identificadas como regras de ratificação das definições da tarefa, viabilizando a identificação de problemas na documentação da tarefa.

8. CONSIDERAÇÕES FINAIS

Neste capítulo são sugeridos trabalhos futuros decorrentes do conhecimento obtido até o momento sobre o tema de *Software Crowdsourcing*, assim como são apresentadas as limitações percebidas durante a realização desta pesquisa.

Ao recapitular as atividades desenvolvidas durante este estudo pode-se perceber que o tema de pesquisa, que trata da percepção dos membros da multidão em relação à documentação das tarefas, é pouco explorado na literatura científica, o que sugere a necessidade de mais pesquisas na área. Este processo de identificação e compreensão destas percepções permite que tanto o meio acadêmico, como a indústria possam ter seus conhecimentos expandidos sobre o modelo de *Software Crowdsourcing*, o que viabiliza o seu aprimoramento de modo eficiente e sustentável.

Sob a perspectiva acadêmica, acredita-se que as informações obtidas pelo estudo possam agregar na consolidação dos conhecimentos relacionados ao *Software Crowdsourcing*, de modo a ampliar e diversificar as áreas abordadas pelo tema e, principalmente, permitir a sua evolução com novas pesquisas. Do ponto de vista relacionado à formação dos alunos, este tipo de conhecimento pode promover uma melhor adequação de expectativas dos alunos frente a este modelo de negócio, como também propiciar eventuais adequações em materiais e abordagens das disciplinas que tratam desta área.

Sob a perspectiva da indústria, considera-se oportuno destacar que os resultados desta investigação podem apoiar no aperfeiçoamento do modelo de trabalho de plataformas que atuam neste nicho de mercado. Os conhecimentos obtidos tornam mais claro como um dos elementos mais importantes deste modelo de negócio, a Multidão, percebe, entende e responde às tarefas propostas pelos contratantes. A compreensão do modo de operação da multidão frente às tarefas propicia que as plataformas possam adequar suas estruturas de modo a obter uma melhor relação entre elas e a multidão. Neste sentido, acredita-se que o modelo de documentação apresentado poderá contribuir para melhorar a qualidade das entregas realizadas na plataforma TopCoder.

8.1 Limitações

Ao realizar uma retrospectiva de todo o processo realizado para esta pesquisa foram identificados determinados fatores que limitaram o estudo e que são apresentados a seguir:

- Avaliou-se como limitação para a realização desta pesquisa o fato de que ela foi desenvolvida por apenas um pesquisador. Mesmo existindo a supervisão da professora orientadora e o apoio dos colegas do grupo de pesquisa na validação e testes dos artefatos criados, a possibilidade de discussão e trabalho com outros pesquisadores interessados e envolvidos com o mesmo tema de pesquisa poderia tornar a pesquisa mais rica e completa.

- Com relação aos participantes que realizaram o Estudo de Caso, estes apresentavam um conjunto heterogêneo de formações e experiências. Esse aspecto propicia percepções diversificadas sobre um mesmo tema e enriquece a pesquisa, entretanto, esta mesma diversidade gerou restrições na seleção das tarefas. Essas restrições estão relacionadas ao fato de que nem todos os participantes possuíam os conhecimentos necessários para a execução de determinadas tarefas.
- Com relação aos participantes que realizaram o Estudo de Campo, estes possuíam a experiência necessária para realizar as atividades, mas a diversidade na formação e experiências foi reduzida uma vez que as tarefas selecionadas abordavam propositalmente as tecnologias Java e Angular, as quais estes profissionais possuíam ampla experiência. Assim, em função deste grupo possuir um perfil relativamente semelhante, o conjunto de percepções obtidas também foi reduzido. Outra questão que envolveu esse grupo foi o tempo necessário para realizar um estudo completo, desde a seleção até a submissão das tarefas, que foi inviabilizado pela falta de tempo necessário à sua execução.

8.2 Trabalhos Futuros

Conforme mencionado, o modelo de negócio proposto pelo *Crowdsourcing* é relativamente novo, e assim necessita ser explorado para que possa evoluir. Nesse sentido, o *Software Crowdsourcing* também deve buscar este mesmo avanço. A partir desta perspectiva, acredita-se que este é um segmento que possui diversas áreas a serem exploradas. Como este é um modelo de negócio aplicado, utilizado pela indústria, sua evolução deveria considerar tanto as necessidades acadêmicas como as necessidades da indústria. Assim, são destacadas abaixo algumas sugestões para pesquisas futuras sobre este tema:

- Dar continuidade a este estudo no sentido de obter um volume mais diversificado em relação aos membros da multidão e assim validar ou não as propostas realizadas nesta pesquisa.
- Viabilizar junto à plataforma TopCoder uma forma de realização de um piloto utilizando o modelo proposto para a documentação da tarefa e assim verificar os resultados e a compreensão da tarefa por parte da multidão, bem como validar a eficácia nos resultados obtidos das submissões destas tarefas.
- Considerando que esta pesquisa foi realizada a partir das informações e estruturas disponíveis na plataforma TopCoder, acredita-se que o modelo proposto para a documentação possa também ser aplicado a outras plataformas que atuem neste nicho de *Software Crowdsourcing*. Isso porque nenhum dos elementos propostos possui uma vinculação exclusiva com a TopCoder, neste sentido a proposta apresenta um caráter genérico.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Allahbakhsh, M.; Benatallah, B.; Ignjatovic, A.; Nezhad, H. R. M.; Bertino, E.; Dustdar, S. "Quality control in crowdsourcing systems: Issues and directions", *IEEE Internet Computing*, vol. 17-2, Mar-Abr 2013, pp. 76-81.
- [2] Bardin, L. "Análise de Conteúdo". Edições 70, 2016, 279p.
- [3] Biolchini, J.; Mian, P. G.; Natali, A. C. C.; Travassos, G. H. "Systematic review in software engineering", Technical report, Systems Engineering and Computer Department, Universidade Federal do Rio de Janeiro, 2005, 31p.
- [4] Brabham, D. C. "Crowdsourcing - The MIT Press essential knowledge series". The MIT Press, 2013, 163p.
- [5] Brooks, F. P. "No silver bullet - essence and accidents of software engineering", *IEEE Computer*, vol. 20-4, Abr 1987, pp. 10-19.
- [6] Creswell, J. W. "Projeto de pesquisa métodos qualitativo, quantitativo e misto". Artmed, 2010, 296p.
- [7] Creswell, J. W. "Investigação Qualitativa e Projeto de Pesquisa : Escolhendo entre Cinco Abordagens". Penso Editora, 2014, 341p.
- [8] Dalpiaz, F.; Korenko, M.; Salay, R.; Chechik, M. "Using the crowds to satisfy unbounded requirements". In: International Workshop on Crowd-Based Requirements Engineering, 2015, pp. 19-24.
- [9] Fu, Y.; Chen, H.; Song, F. "STWM: A solution to self-adaptive task-worker matching in software crowdsourcing". In: International Conference of Algorithms and Architectures for Parallel Processing, 2015, pp. 383-398.
- [10] Gil, A. C. "Estudo de caso". Atlas, 2009, 168p.
- [11] Green, A. E.; Hoyos, M. D.; Barnes, S.-A.; Baldauf, B.; Behle, H. "Crowdemploy crowdsourcing case studies: An empirical investigation into the impact of crowdsourcing on employability", Technical report, Publications Office of the European Union, 2013, 108p.
- [12] Groen, E. C. "Crowd out the competition". In: International Workshop on Crowd-Based Requirements Engineering, 2015, pp. 13-18.
- [13] Groen, E. C.; Seyff, N.; Ali, R.; Dalpiaz, F.; Dörr, J.; Guzman, E.; Hosseini, M.; Marco, J.; Oriol, M.; Perini, A.; Stade, M. J. C. "The crowd in requirements engineering: The landscape and challenges", *IEEE Software*, vol. 34-2, Mar-Abr 2017, pp. 44-52.

- [14] Hosseini, M.; Phalp, K.; Taylor, J.; Ali, R. "The four pillars of crowdsourcing: A reference model". In: International Conference on Research Challenges in Information Science, 2014, pp. 1–12.
- [15] Hosseini, M.; Shahri, A.; Phalp, K.; Taylor, J.; Ali, R. "Crowdsourcing: A taxonomy and systematic mapping study", *Computer Science Review*, vol. 17, Mai 2015, pp. 43–69.
- [16] Howe, J. "Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business". The Crown Publishing Group, 2008, 322p.
- [17] IIBA, I. I. o. B. A. "A Guide to the Business Analysis Body of Knowledge (BABOK), Version 2.0". International Institute of Business Analysis, 2009, 271p.
- [18] IIBA, I. I. o. B. A. "Guide to the Business Analysis Body of Knowledge (BABOK) V3". International Institute of Business Analysis, 2015, 514p.
- [19] ISO. "Iso/iec/ieee 29148:2011: Systems and software engineering — life cycle processes — requirements engineering". Recuperado de: <https://www.iso.org/home.html>, Dezembro 2011.
- [20] Kanchev, G. M.; Chopra, A. K. "Social media through the requirements lens: A case study of google maps". In: International Workshop on Crowd-Based Requirements Engineering, 2015, pp. 7–12.
- [21] Kitchenham, B. "Procedures for performing systematic reviews", Technical report, Software Engineering Group Department of Computer Science, Keele University, 2004, 33p.
- [22] Lakhani, K.; Garvin, D.; Lonstein, E. "Topcoder (a): Developing software through crowdsourcing", *Harvard Business School General Management*, vol. 610–032, Jan 2010, pp. 1–19.
- [23] Lebraty, J.-F.; Lobre-Lebraty, K. "Crowdsourcing: One step beyond". John Wiley & Sons, 2013, 137p.
- [24] Levy, M.; Hadar, I.; Te'eni, D. "A gradual approach to crowd-based requirements engineering: The case of conference online social networks". In: International Workshop on Crowd-Based Requirements Engineering, 2015, pp. 25–30.
- [25] Machado, L.; Zanatta, A.; Marczak, S.; Prikładnicki, R. "The good, the bad and the ugly: an onboard journey in software crowdsourcing competitive model". In: International Workshop on CrowdSourcing in Software Engineering, 2017, pp. 2–8.
- [26] Mao, K.; Capra, L.; Harman, M.; Jia, Y. "A survey of the use of crowdsourcing in software engineering", Technical report, UCL Department of Computer Science, University College London, 2015, 36p.

- [27] Mao, K.; Capra, L.; Harman, M.; Jia, Y. "A survey of the use of crowdsourcing in software engineering", *Journal of Systems and Software*, vol. 126, Abr 2017, pp. 57–84.
- [28] Mao, K.; Yang, Y.; Wang, Q.; Jia, Y.; Harman, M. "Developer recommendation for crowdsourced software development tasks". In: *IEEE Symposium on Service-Oriented System Engineering*, 2015, pp. 347–356.
- [29] Mcgrath, J. E. "Methodology matters: Doing research in the behavioral and social sciences". Elsevier, 1995, cap. Part II: The Process of Developing Interactive Systems, pp. 152–169.
- [30] Prikladnicki, R.; Machado, L.; Carmel, E.; de Souza, C. R. B. "Brazil software crowdsourcing: a first step in a multi-year study". In: *International Workshop on CrowdSourcing in Software Engineering*, 2014, pp. 1–4.
- [31] Ramakrishnan, S.; Srinivasaraghavan, V. "Delivering software projects using captive university crowd". In: *International Workshop on Cooperative and Human Aspects of Software Engineering*, 2014, pp. 115–118.
- [32] Saxton, G. D.; Oh, O.; Kishore, R. "Rules of crowdsourcing: Models, issues, and systems of control", *Information Systems Management*, vol. 30–1, Jan 2013, pp. 2–20.
- [33] Shao, W.; Wang, X.; Jiao, W. "A Developer Recommendation Framework in Software Crowdsourcing Development". Springer, 2016, cap. Framework and System, pp. 151–164.
- [34] Snijders, R.; Dalpiaz, F.; Brinkkemper, S.; Hosseini, M.; Ali, R.; Ozum, A. "Refine: A gamified platform for participatory requirements engineering". In: *International Workshop on Crowd-Based Requirements Engineering*, 2015, pp. 1–6.
- [35] Sommerville, I. "Engenharia de Software". Pearson, 2007, 552p.
- [36] Srivastava, P. K.; Sharma, R. "Crowdsourcing to elicit requirements for myerp application". In: *International Workshop on Crowd-Based Requirements Engineering*, 2015, pp. 31–35.
- [37] Stol, K.; Fitzgerald, B. "Two's company, three's a crowd: a case study of crowdsourcing software development". In: *International Conference on Software Engineering*, 2014, pp. 187–198.
- [38] Vaz, L. "Software crowdsourcing: Revisão da literatura sobre a documentação das tarefas", Monografia, Escola Politécnica, Programa de Pós-Graduação em Ciência da Computação, PUCRS, 2017, 65p.
- [39] Wang, H.; Yin, G.; Li, X.; Li, X. "TRUSTIE: a software development platform for crowdsourcing". Springer, 2015, cap. Software Crowdsourcing Systems, pp. 165–190.
- [40] Wieggers, K. "Software Requirements". Microsoft Press, 2013, 673p.

- [41] Wohlin, C. "Guidelines for snowballing in systematic literature studies and a replication in software engineering". In: International Conference on Evaluation and Assessment in Software Engineering, 2014, pp. 1–10.
- [42] Wu, W.; Tsai, W.; Li, W. "An evaluation framework for software crowdsourcing", *Frontiers of Computer Science*, vol. 7–5, Out 2013, pp. 694–700.
- [43] Yin, R. K. "Estudo de Caso: Planejamento e Métodos". Bookman editora, 2015, 320p.
- [44] Zanatta, A. L.; Machado, L.; Pereira, G.; Prikladnicki, R.; Carmel, E. "Software crowdsourcing platforms", *IEEE Software*, vol. 33–6, Nov-Dez 2016, pp. 112–116.
- [45] Zhu, J.; Shen, B.; Hu, F. "A learning to rank framework for developer recommendation in software crowdsourcing". In: Asia-Pacific Software Engineering Conference, 2015, pp. 285–292.

APÊNDICE A – ESTUDO DE CASO: FORMULÁRIO PERFIL DO PARTICIPANTE



Pontifícia Universidade Católica do Rio Grande do Sul
 Faculdade de Informática – PPGCC
 Desenvolvimento Colaborativo de Software – Prof^a. Sabrina Marczak

Questionário – Perfil do Aluno

► Identificação

Dados dos Alunos: *	
Nome Completo:	
E-mail ACAD:	
Curso de Formação: (Graduação)	
Tempo de Experiência em TI: (em anos)	

Se você já possui cadastro na plataforma TopCoder informe seu usuário de login:

► Dados Gerais

1. Para os perfis abaixo indique sua experiência: *

(Caso você não possua experiência em um dos itens do perfil informe zero nos campos do formulário)

Grupo de Atividade	Experiência Acadêmica (em anos)	Experiência Profissional (em anos)
Análise (Requisitos, Negócio, Processos, etc.)		
Desenvolvimento (Programador, Arquiteto, etc.)		
Design (UX Design, Projeto de Interface, etc.)		
Gestão (Gerente de Projeto, Product Owner, Scrum Master, etc.)		
Teste (Análise, Teste Manual e/ou Automatizado, etc.)		



Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática – PPGCC
Desenvolvimento Colaborativo de Software – Prof^a. Sabrina Marczak

2. Caso a questão anterior não apresente um perfil que você já atuou, informe abaixo qual é este perfil e o seu tempo de experiência:

3. Para os perfis abaixo indique sua experiência: *

	Tenho domínio avançado	Conheço muito bem	Sei utilizar	Conheço pouco	Não sei utilizar
.NET (ASP, C#)					
Angular					
Android					
Bootstrap					
C					
C++					
CSS					
Docker					
Groovy					
HTML5					
Ionic					
Java					
JavaScript					
JSON					
JSP					
MySQL					
Node					
Oracle 10g					
Python					
ReactJS					



Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática – PPGCC
Desenvolvimento Colaborativo de Software – Prof^a. Sabrina Marczak

4. Caso a questão anterior não apresente linguagens que você possui experiência informe abaixo quais são estas e o nível de conhecimento:

5. Descreva brevemente sua experiência indicando as principais atividades que você desempenhou: *
 (Por exemplo: Atuei como Desenvolvedor Sênior JAVA em um projeto de ERP, depois nesse mesmo projeto passei a atuar como arquiteto de integrações.)

► **Crowdsourcing**

A partir do conceito apresentado abaixo responda as questões que seguem:

Considerando o conceito amplo, podemos descrever **Crowdsourcing** como um modelo de negócio que permite a cooperação de um grande conjunto de pessoas (multidão) a fim de chegar a um objetivo comum. As atividades realizadas por esta multidão podem envolver a coleta de informações, execução de tarefas específicas ou mesmo fomentar financiamentos.

6. Você já contribuiu para alguma iniciativa de **Crowdsourcing**? *
- (Por exemplo: indicar sobre algum buraco existente em uma via pública no WAZE, revisar algum documento na WIKIPEDIA ou realizar confirmação da existência de um estabelecimento no Google Maps)

() Sim

() Não

7. Se você respondeu 'Sim' na Questão 6, responda a presente questão.

Descreva brevemente o(s) tipo(s) iniciativa(s) de **Crowdsourcing** que você participou e qual(is) tipo de ações você realizou. *

(Por exemplo: WAZE, WIKIPEDIA ou GOOGLE MAPS)



Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática – PPGCC
Desenvolvimento Colaborativo de Software – Prof^a. Sabrina Marczak

► **Software Crowdsourcing**

A partir do conceito apresentado abaixo responda as questões que seguem:

Podemos definir de forma simplificada, **Software Crowdsourcing** como uma ação em que uma empresa ou instituição decide terceirizar uma tarefa de maior complexidade para uma rede indefinida (e geralmente grande) de pessoas na forma de uma chamada aberta. Isso pode assumir a forma de um sistema cooperativo (quando a tarefa é realizada colaborativa), mas também é frequentemente realizado por pessoas individualmente. O pré-requisito principal é o formato de chamada aberta a ampla para uma rede de potenciais de colaboradores capacitados.

8. Você possui experiência em projetos de *Software Crowdsourcing*? *

() Sim

() Não

9. Se você respondeu 'Sim' na Questão 8, responda a presente questão.

Informe o tempo de experiência que você possui em projetos de *Software Crowdsourcing*? *

(Informe sua experiência em anos)

10. Como você tomou conhecimento das oportunidades de contribuir em projetos de *Software Crowdsourcing*? *

11. O que motivou você a buscar estas oportunidades em projetos de *Software Crowdsourcing*? *

12. Cite as plataformas em que você realizou atividades de projetos de *Software Crowdsourcing*? *

(por exemplo: Amazon Mechanical Turk, TopCoder, uTeste, etc)



Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática – PPGCC
Desenvolvimento Colaborativo de Software – Prof^a. Sabrina Marczak

13. Cite quais os tipos de tarefas que você realizou em projetos de *Software Crowdsourcing*? *
(Exemplo: Codificação, Design, Teste, UX, etc.)

14. Caso você tenha participado de mais de uma tarefa em projetos de *Software Crowdsourcing*, o que o levou a procurar por outras tarefas? Caso não tenha continuado, o que o desmotivou/impediu de continuar? *

15. Se você já trabalhou com *Software Crowdsourcing* anteriormente, o que o manteve engajado e o levou a continuar participando? *

16. De forma geral, como você descreve sua experiência em projetos de *Software Crowdsourcing*? *

APÊNDICE B – ESTUDO DE CASO: FORMULÁRIO SELEÇÃO DA TAREFA (CICLO 1)



Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática – PPGCC
Desenvolvimento Colaborativo de Software – Prof^a. Sabrina Marczak

Questionário – Seleção da Tarefa 1

► Identificação

Identificação do aluno *	
Nome Completo:	
Login na TopCoder:	

► Informações da Tarefa

1. Informe o nome da tarefa que você selecionou na plataforma TopCoder e o seu tipo (*Subtrack*). *

Exemplo: Tipo: CODE / Tarefa: Hercules Retail 360 iOS App – Timetrade API integration

Tipo(<i>Subtrack</i>)	Nome da Tarefa

2. A partir da leitura do enunciado e documentação da tarefa que você selecionou, descreva com suas palavras qual o objetivo da mesma. *

3. Descreva resumidamente o planejamento que você pretende realizar para entregar a tarefa selecionada. *



Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática – PPGCC
Desenvolvimento Colaborativo de Software – Prof^a. Sabrina Marczak

► **Critérios de Seleção da Tarefa**

4. Descreva quais foram os critérios que você utilizou para selecionar a sua tarefa. *

5. Descreva pontos relevantes que fizeram você descartar as demais tarefas oferecidas pela plataforma. *

APÊNDICE C – ESTUDO DE CASO: DIÁRIO DE BORDO



Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática – PPGCC
Desenvolvimento Colaborativo de Software – Prof^a. Sabrina Marczak

Diário de Bordo

Nome Completo

Anotações do Diário

APÊNDICE D – ESTUDO DE CASO: FORMULÁRIO AVALIAÇÃO DA TAREFA (CICLO 1)



Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática – PPGCC
Desenvolvimento Colaborativo de Software – Prof^a. Sabrina Marczak

Questionário – Entrega da Tarefa 1

► Instruções de preenchimento

Responda às questões a seguir sob a perspectiva da tarefa que você selecionou independente se você conseguiu finalizá-la ou não.

Observações:

- Caso você tenha realizado mais de uma tarefa, responda o questionário tantas vezes quanto o número de tarefas realizadas.
- As questões indicadas com * são obrigatórias;

► Identificação do Aluno *

Nome Completo	
Usuário no TopCoder	
Nome da Tarefa (utilize o nome conforme apresentado na plataforma TopCoder)	

► Dados Gerais

1. Como você classifica a sua participação na realização da tarefa de *Software Crowdsourcing* na plataforma TopCoder? *

Muito satisfeito	Satisfeito	Regular	Pouco satisfeito	Insatisfeito
()	()	()	()	()

2. Quais aspectos você destaca na sua experiência em realizar esta tarefa utilizando a plataforma TopCoder:

- a. Aspectos Positivos *

--



Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática – PPGCC
Desenvolvimento Colaborativo de Software – Prof^a. Sabrina Marczak

b. Aspectos Negativos *

3. Você entregou sua tarefa na plataforma TopCoder? *

- () Sim
 () Não

4. Caso você tenha respondido “Não” na questão 3, responda a presente questão.

O que levou você a não a entregar a tarefa que você selecionou? *

5. Quanto tempo você utilizou para realizar a sua tarefa, tendo entregue ou não a mesma? *

(indique o tempo em horas - preencha apenas com valores numéricos)

► **Dados da Tarefa**

Avalie as questões que seguem considerando toda a documentação fornecida para a tarefa.

Por documentação da tarefa se entende:

- **a descrição da tarefa;** e/ou
- **a documentação complementar** (especificações de regras de negócio ou técnicas, padrões de desenvolvimento, diagramas, critérios de aceitação da tarefa, etc).

6. Como você classifica a documentação da tarefa que você selecionou na plataforma TopCoder? *

Excelente	Muito boa	boa	Fraca	Pobre
()	()	()	()	()



Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática – PPGCC
Desenvolvimento Colaborativo de Software – Prof^a. Sabrina Marczak

7. Como você avalia a documentação que você utilizou para realizar a(s) tarefa(s)? Opine considerando como cada um dos critérios a seguir se relaciona com a documentação utilizada. *

Critério	Descrição	Classificação
Atômico	A documentação deve ser entendida independentemente de outras tarefas co-relacionadas, deve ser autocontida em sua descrição.	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo parcialmente <input type="checkbox"/> Não concordo nem discordo <input type="checkbox"/> Discordo parcialmente <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não se aplica
Completo	A documentação deve possuir um nível de detalhe adequado em sua descrição, isto é, ser suficiente para orientar o trabalho a ser realizado.	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo parcialmente <input type="checkbox"/> Não concordo nem discordo <input type="checkbox"/> Discordo parcialmente <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não se aplica
Preciso	A documentação deve descrever com precisão sua(s) funcionalidade(s) com o propósito de viabilizar a sua construção e validação.	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo parcialmente <input type="checkbox"/> Não concordo nem discordo <input type="checkbox"/> Discordo parcialmente <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não se aplica
Coesão	Uma documentação coesa consiste em um conjunto de informações que estão relacionadas a apenas uma coisa, seja ela um processo de negócio, regra de negócio, unidade organizacional, etc. Todos os documentos devem dar suporte e apoiar o escopo geral da atividade.	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo parcialmente <input type="checkbox"/> Não concordo nem discordo <input type="checkbox"/> Discordo parcialmente <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não se aplica
Consistente	A documentação deve estar alinhada com as necessidades identificadas pelas partes interessadas e não apresentar conflito com outras tarefas.	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo parcialmente <input type="checkbox"/> Não concordo nem discordo <input type="checkbox"/> Discordo parcialmente <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não se aplica
Viabilidade	A documentação deve descrever um atividade factível considerando os riscos, cronograma e orçamento acordados, ou ser considerado viável o suficiente para uma investigação através de experimentos ou protótipos.	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo parcialmente <input type="checkbox"/> Não concordo nem discordo <input type="checkbox"/> Discordo parcialmente <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não se aplica
Conciso	Não deve conter conteúdo desnecessário na documentação da tarefa.	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo parcialmente <input type="checkbox"/> Não concordo nem discordo <input type="checkbox"/> Discordo parcialmente <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não se aplica
Necessário	Cada tarefa deve descrever (documentar) uma funcionalidade que agregue valor de negócios.	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo parcialmente <input type="checkbox"/> Não concordo nem discordo <input type="checkbox"/> Discordo parcialmente <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não se aplica
Ajustabilidade/adaptação	Tarefas inter-relacionadas devem ser mantidas agrupadas a fim de se tornem modificáveis. Essa característica é demonstrada através de uma estruturação lógica dos requisitos.	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo parcialmente <input type="checkbox"/> Não concordo nem discordo <input type="checkbox"/> Discordo parcialmente <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não se aplica
Não Ambiguidade	A documentação de uma tarefa nunca deve ser dúbia, ou seja, a documentação não pode permitir a formação de múltiplas e divergentes interpretações válidas.	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo parcialmente <input type="checkbox"/> Não concordo nem discordo <input type="checkbox"/> Discordo parcialmente <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não se aplica
Testável	A documentação deve prover mecanismos que permitam a sua verificação com o objetivo de avaliar se a atividade foi cumprida. Os níveis aceitáveis de verificação dependem do nível de abstração proposto.	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo parcialmente <input type="checkbox"/> Não concordo nem discordo <input type="checkbox"/> Discordo parcialmente <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não se aplica



Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática – PPGCC
Desenvolvimento Colaborativo de Software – Prof^a. Sabrina Marczak

Critério	Descrição	Classificação
Priorizado	A tarefa deve estar devidamente classificada e negociado em termos de importância e valor em relação a todas as demais tarefas propostas pelo cliente.	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo parcialmente <input type="checkbox"/> Não concordo nem discordo <input type="checkbox"/> Discordo parcialmente <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não se aplica
Correção	Defeitos existentes na documentação da tarefa podem levar a defeitos na solução resultante.	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo parcialmente <input type="checkbox"/> Não concordo nem discordo <input type="checkbox"/> Discordo parcialmente <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não se aplica

8. A tarefa que você realizou possuía documentação complementar além da que é apresentada na sua descrição? *

(links para outros documentos ou documentos disponíveis no fórum)

- Sim
 Não

9. Caso você tenha respondido "Sim" na questão 8, responda as questões 9 e 10.

Como você classifica a documentação complementar disponível para a tarefa que você selecionou na plataforma TopCoder? *

Excelente	Muito boa	boa	Fraca	Pobre
<input type="checkbox"/>				

10. Foi necessário solicitar algum tipo de esclarecimento à plataforma TopCoder para compreender o objetivo da tarefa que você selecionou? *

- Sim
 Não

11. Caso você tenha respondido "Sim" na questão 10, responda as questões 11 e 12.

Qual o canal de comunicação você utilizou para entrar em contato com a plataforma para solicitar os esclarecimentos? *

- Fórum
 E-mail
 Outro (cite) _____

12. Você obteve retorno dos esclarecimentos solicitados à plataforma TopCoder sobre a tarefa? *

- Sim
 Não



Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática – PPGCC
Desenvolvimento Colaborativo de Software – Prof^a. Sabrina Marczak

13. **Caso você tenha respondido "Sim" na questão 10 e na questão 12.**

Como você classifica os esclarecimentos recebidos para prosseguir com sua tarefa? *

Excelente	Muito boa	boa	Fraca	Pobre
()	()	()	()	()

14. Foi necessário buscar algum tipo de informação adicional, fora da plataforma TopCoder, para compreender o objetivo da tarefa que você selecionou? *

() Sim

() Não

15. **Se você respondeu 'Sim' na Questão 14, responda a presente questão.**

Que tipo de informação adicional você buscou em fontes externas à plataforma TopCoder para compreender o objetivo da tarefa selecionada? *

16. Qual a sua percepção ao realizar uma tarefa que você, como membro da multidão (*crowdworker*), não possui o contexto geral na qual esta tarefa está inserida? *

17. Em sua opinião, quais pontos podem ser melhorados em relação a descrição da documentação (enunciado da tarefa e documentação complementar) associada à tarefa de forma a contribuir para a sua realização? *

APÊNDICE E – ESTUDO DE CASO: FORMULÁRIO SELEÇÃO DA TAREFA (CICLO 2)



Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática – PPGCC
Desenvolvimento Colaborativo de Software – Prof^a. Sabrina Marczak

Questionário – Seleção da Tarefa 2

► Identificação

Identificação do aluno	
Nome Completo:	

► Informações da Tarefa

1. Informe o nome da tarefa que você selecionou na plataforma TopCoder e o seu tipo (*Subtrack*). *

Exemplo: Tipo: CODE / Nome da Tarefa: Hercules Retail 360 iOS App – Timetrade API integration

Tipo(<i>Subtrack</i>)	Nome da Tarefa

2. A partir da leitura do enunciado e da documentação da tarefa que você selecionou, descreva com suas palavras qual o objetivo da mesma. *



Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática – PPGCC
Desenvolvimento Colaborativo de Software – Prof^a. Sabrina Marczak

► **Critérios de Seleção da Tarefa**

3. Descreva quais foram os fatores que influenciaram na seleção da sua tarefa. *

4. Descreva quais foram os fatores que influenciaram você desconsiderar outras tarefas. *

APÊNDICE F – ESTUDO DE CASO: FORMULÁRIO AVALIAÇÃO DA TAREFA (CICLO 2)



Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática – PPGCC
Desenvolvimento Colaborativo de Software – Prof^a. Sabrina Marczak

Questionário – Entrega da Tarefa 2

► Instruções de preenchimento

Responda às questões a seguir considerando sua experiência na realização da(s) tarefa(s) na plataforma TopCoder, independente se você conseguiu finalizá-la e submetê-la ou não.

Observações:

- Caso você tenha realizado mais de uma tarefa, responda o questionário tantas vezes quanto o número de tarefas realizadas.
- As questões indicadas com * são obrigatórias;

► Identificação do Aluno *

Nome Completo	
Nome da Tarefa (utilize o nome conforme apresentado na plataforma TopCoder)	

► Dados Gerais

1. Como você classifica a sua participação na realização da tarefa da plataforma TopCoder? *

Insatisfeito	Pouco satisfeito	Regular	Satisfeito	Muito satisfeito
()	()	()	()	()

2. Você entregou sua tarefa na plataforma TopCoder? *

- () Sim
() Não

3. Caso você tenha respondido "Não" na questão 2, responda a presente questão.
O que levou você a não a entregar a tarefa que você selecionou? *

4. Quanto tempo você utilizou para realizar a sua tarefa, tendo entregue ou não a mesma? *
(indique o tempo em horas - preencha apenas com valores numéricos)



Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática – PPGCC
Desenvolvimento Colaborativo de Software – Profª. Sabrina Marczak

► **Dados da Tarefa**

Avalie as questões que seguem considerando toda a documentação fornecida para a tarefa.

Por documentação da tarefa se entende:

- **a descrição da tarefa;** e/ou
- **a documentação complementar**
 (especificações de regras de negócio ou técnicas, padrões de desenvolvimento, diagramas, critérios de aceitação da tarefa, etc).

5. Como você classifica a documentação da tarefa que você selecionou na plataforma TopCoder? *

Fraca	Pobre	boa	Muito boa	Excelente
()	()	()	()	()

6. Como você avalia a documentação que você utilizou para realizar a tarefa? *

Opine considerando como cada um dos critérios a seguir se relaciona com a documentação utilizada.

ID	Descrição	Classificação
1	A documentação fornecida para a sua tarefa estava devidamente autocontida , ou seja, não necessita de outras informações para executar a tarefa.	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo parcialmente <input type="checkbox"/> Não concordo nem discordo <input type="checkbox"/> Discordo parcialmente <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não se aplica
2	A documentação fornecida para a sua tarefa possui o nível de detalhamento necessário para executar a tarefa, ou seja, descreve as etapas que devem ser realizadas de forma satisfatória para a finalização da tarefa.	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo parcialmente <input type="checkbox"/> Não concordo nem discordo <input type="checkbox"/> Discordo parcialmente <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não se aplica
3	A documentação fornecida para a sua tarefa possui o nível de precisão necessária para executar a tarefa, ou seja, descreve os detalhes de como os procedimentos devem ser realizados para finalizar a tarefa.	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo parcialmente <input type="checkbox"/> Não concordo nem discordo <input type="checkbox"/> Discordo parcialmente <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não se aplica
4	A documentação fornecida para a sua tarefa está corretamente documentada e coesa de forma a dar o suporte necessário para executar a tarefa.	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo parcialmente <input type="checkbox"/> Não concordo nem discordo <input type="checkbox"/> Discordo parcialmente <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não se aplica
5	A documentação fornecida para a sua tarefa descreve uma tarefa factível dentro do prazo e orçamento proposto.	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo parcialmente <input type="checkbox"/> Não concordo nem discordo <input type="checkbox"/> Discordo parcialmente <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não se aplica



Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática – PPGCC
Desenvolvimento Colaborativo de Software – Profª. Sabrina Marczak

ID	Descrição	Classificação
6	A documentação fornecida para a sua tarefa não possui informações desnecessárias para a execução da tarefa.	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo parcialmente <input type="checkbox"/> Não concordo nem discordo <input type="checkbox"/> Discordo parcialmente <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não se aplica
7	A documentação fornecida para a sua tarefa possui informações dúbias , ou seja, pode apresentar interpretações divergentes conforme o interlocutor que estiver lendo esta documentação.	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo parcialmente <input type="checkbox"/> Não concordo nem discordo <input type="checkbox"/> Discordo parcialmente <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não se aplica
8	A documentação fornecida para a sua tarefa possui informações suficientes para realizar os testes necessários a fim de validar se a tarefa foi cumprida conforme esperado.	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo parcialmente <input type="checkbox"/> Não concordo nem discordo <input type="checkbox"/> Discordo parcialmente <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não se aplica
9	A documentação fornecida para a sua tarefa possui problemas que podem gerar defeitos no resultado final da tarefa.	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo parcialmente <input type="checkbox"/> Não concordo nem discordo <input type="checkbox"/> Discordo parcialmente <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não se aplica

7. A tarefa que você realizou possuía documentação complementar além da que é apresentada na sua descrição? *

(links para outros documentos ou documentos disponíveis no fórum)?

- Sim
 Não

8. Caso você tenha respondido "Sim" na questão 7, responda a seguinte questão.

Como você classifica a documentação complementar disponível para a tarefa que você selecionou na plataforma TopCoder?

Fraca	Pobre	boa	Muito boa	Excelente
<input type="checkbox"/>				

9. Foi necessário solicitar algum tipo de esclarecimento à plataforma TopCoder para compreender o objetivo da tarefa que você selecionou? *

- Sim
 Não

10. Caso você tenha respondido "Sim" na questão 9, responda a seguinte questão.

Como você classifica os esclarecimentos recebidos para prosseguir com sua tarefa? *

Fraca	Pobre	boa	Muito boa	Excelente
<input type="checkbox"/>				



Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática – PPGCC
Desenvolvimento Colaborativo de Software – Profª. Sabrina Marczak

11. Em sua opinião, quais pontos podem ser melhorados em relação a descrição da documentação (enunciado da tarefa e documentação complementar) associada à tarefa de forma a contribuir para a sua realização. *

► **Motivação**

12. Em linhas gerais, como você avalia os itens a seguir em função de sua experiência no desenvolvimento das tarefas desenvolvidas na TopCoder. *

	Discordo fortemente	Discordo parcialmente	Não discordo nem concordo	Concordo parcialmente	Concordo fortemente
1. Como a plataforma está em inglês, tive dificuldades para compreender a tarefa.	()	()	()	()	()
2. Tive dificuldade em encontrar uma tarefa para executar de acordo com minhas habilidades ou perfil.	()	()	()	()	()
3. Mesmo encontrando uma tarefa de acordo com meu perfil ou habilidades, tive dificuldades para compreender a descrição da tarefa.	()	()	()	()	()
4. Tive problemas para configurar o ambiente necessário para realizar a tarefa selecionada.	()	()	()	()	()
5. Mesmo encontrando uma tarefa de acordo com meu perfil ou habilidades, tive dificuldades para executá-la.	()	()	()	()	()
6. Ao realizar a tarefa, evitei a interação com os demais participantes da plataforma (por exemplo, através do fórum), pois eles se comunicam apenas em inglês.	()	()	()	()	()
7. Tive dificuldades na interação com os demais membros da plataforma ao realizar a tarefa.	()	()	()	()	()
8. Tive dificuldade de gerenciar a alocação de tempo disponível para executar a tarefa.	()	()	()	()	()

13. Caso você tenha respondido "Discordo fortemente" ou "Discordo parcialmente" na questão 12, responda a questão seguinte.

Como na questão anterior você indicou que teve dificuldade em gerenciar o tempo para executar a tarefa, descreva quais foram estas dificuldades, comente sobre elas. *



Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática – PPGCC
Desenvolvimento Colaborativo de Software – Profª. Sabrina Marczak

14. Considerando sua experiência na realização da(s) tarefas(s) na plataforma TopCoder e eventuais barreiras encontradas durante o processo, qual a probabilidade de você continuar a participar de tarefas futuras nesta plataforma? *

- Muito improvável. Não pretendo participar de mais nenhuma tarefa na plataforma.
- Pouco provável. Por alguns momentos, fiquei desmotivado a seguir em frente, mas pretendo participar de futuras tarefas, na esperança de que não aconteça de novo.
- Não tenho certeza. Não cheguei a me deparar com algo que me incomodasse ao ponto de deixar de participar. Ainda pretendo participar, na medida do possível, de tarefas na plataforma.
- Provável. Encontrei pontos em que a plataforma e/ou o processo poderia ser melhor, mas não me impediu de realizar a(s) tarefa(s). Pretendo seguir participando da plataforma.
- Muito provável. Não encontrei algo que me deixasse desmotivado a continuar. Acredito que sejam situações comuns, que podem acontecer em qualquer outro(a) lugar/plataforma. Seguirei participando de tarefas na plataforma.

15. Considerando eventuais pontos positivos ou negativos que você tenha identificado em sua experiência de ter participado das tarefas na plataforma TopCoder, qual o seu nível de satisfação? *

- Muito insatisfeito(a). Não encontrei pontos positivos, somente negativos.
- Insatisfeito(a). Encontrei mais pontos negativos do que positivos ou a relevância dos pontos positivos é muito inferior à dos pontos negativos.
- Não tenho certeza. Não encontrei pontos positivos ou negativos
- Pouco satisfeito(a). Encontrei mais pontos positivos do que negativos ou a relevância dos pontos negativos é muito inferior à dos pontos positivos.
- Muito satisfeito(a). Não encontrei pontos negativos, somente positivos.



Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática – PPGCC
Desenvolvimento Colaborativo de Software – Profª. Sabrina Marczak

16. Considerando o esforço e o tempo investidos na realização da(s) tarefa(s), você concorda que o modelo competitivo implementado na plataforma TopCoder é um modelo viável para os participantes, mesmo sabendo que você pode nunca ganhar uma competição? *

- () Discordo plenamente. Não vejo motivos pelos quais alguém participaria de tarefas sob este modelo.
- () Discordo parcialmente. Apesar de não achar viável para mim em função da minha realidade, consigo entender porque algumas pessoas participam deste modelo.
- () Não concordo nem discordo. Não possuo opinião formada a respeito.
- () Concordo parcialmente. Para mim é viável em função da minha disponibilidade e perseverança, mas entendo porque algumas pessoas não seguem participando por muito tempo.
- () Concordo totalmente. Acredito que este seja um modelo perfeitamente viável e entendo que qualquer pessoa consiga participar das tarefas na plataforma.

APÊNDICE G – ESTUDO DE CAMPO: TERMO DE CONFIDENCIALIDADE



Pontifícia Universidade Católica do Rio Grande do Sul – PUCRS
Escola Politécnica
Programa de Pós-Graduação em Ciências da Computação

TERMO DE CONFIDENCIALIDADE

A PUCRS, através das equipes de Engenharia de Software da Escola Politécnica, agradece a todos os participantes a inestimável contribuição que prestam para o avanço da pesquisa aplicada que envolve a Engenharia de Requisitos.

O objetivo desta pesquisa é identificar como a documentação de tarefas (requisitos de software) propostas em plataformas de *Software Crowdsourcing* influenciam na seleção e execução destas tarefas. A pesquisa se baseia em tarefas reais propostas na plataforma TopCoder. Para compreender estas questões, você será convidado a analisar um conjunto de tarefas e responder a um questionário (*survey*) discorrendo sobre a sua percepção em relação a qualidade da documentação, pontos fortes e fracos desta documentação, problemas identificados dentre outros. Espera-se que com o resultado obtido a partir deste estudo seja possível contribuir com a pesquisa na área de *Software Crowdsourcing*.

Lembramos que o objetivo deste estudo **não é** avaliar o participante, **mas sim** os aspectos inerentes ao entendimento de requisitos de software e os impactos decorrentes da qualidade destes documentos. O uso que se faz dos registros efetuados durante a coleta de dados é **estritamente limitado** a atividades de pesquisa e desenvolvimento, garantindo-se para tanto que:

1. O anonimato dos participantes será preservado em todo e qualquer documento divulgado em foros científicos (tais como conferências, periódicos, livros e assemelhados) ou pedagógicos (tais como apostilas de cursos, slides de apresentações, e assemelhados).
2. Todo participante terá acesso a cópias destes documentos após a publicação dos mesmos.
3. Todo participante que se sentir constrangido ou incomodado durante uma situação de teste pode interromper o teste e estará fazendo um favor à equipe se registrar por escrito as razões ou sensações que o levaram a esta atitude. A equipe fica obrigada a descartar o teste para fins da avaliação a que se destinaria.
4. Os participantes que forem menores de idade terão, obrigatoriamente, que apresentar o consentimento de seu responsável, para participação no estudo, o qual será declarado ciente do estudo a ser realizado através de sua assinatura no presente Termo de Consentimento.
5. Todo participante tem direito de expressar por escrito, na data do teste, qualquer restrição ou condição adicional que lhe pareça aplicar-se aos itens acima enumerados (1, 2, 3 e 4). A equipe se compromete a observá-las com rigor e entende que, na ausência de tal manifestação, o participante concorda que rejam o comportamento ético da equipe somente as condições impressas no presente documento.
6. A equipe tem direito de utilizar os dados dos testes, mantidas as condições acima mencionadas, para quaisquer fins acadêmicos, pedagógicos e/ou de desenvolvimento contemplados por seus membros.

[a ser preenchido pelo pesquisador]

Forma: _____ Data: ____ / ____ / ____

Condições especiais (caso não haja condições especiais, escreva “nenhuma”):

continua no verso

Por favor, indique sua posição em relação aos termos acima:

Estou de pleno acordo com os termos acima.

Em anexo registro condições adicionais para este teste.

Assinatura do participante

Assinatura do responsável
(caso o participante seja menor de idade)

Assinatura do pesquisador

Nome do Participante:

Pesquisador Responsável: Escola Politécnica – PPGCC – Luis Fernandes Vaz

APÊNDICE H – ESTUDO DE CAMPO: FORMULÁRIO AVALIAÇÃO DA TAREFA COM PROFISSIONAIS



Pontifícia Universidade Católica do Rio Grande do Sul – PUCRS
Escola Politécnica
Programa de Pós-Graduação em Ciências da Computação

Questionário – Perfil Profissional

► Identificação *

Nome Completo:	
----------------	--

► Experiência

1. Para os perfis abaixo indique sua experiência profissional (em anos): *

(Caso você não possua experiência em um dos itens do perfil informe zero nos campos do formulário)

Análise (Requisitos, Negócio, Processos, etc.)	Desenvolvimento (Programador, Arquiteto, etc.)	Design (UX Design, Projeto de Interface, etc.)	Gestão (Gerente de Projeto, Product Owner, Scrum Master, etc.)	Teste (Análise, Teste Manual e/ou Automatizado, etc.)

2. Como você classifica seu nível de conhecimento para as tecnologias indicadas a seguir: *

JAVA		Angular	
Sênior	()	Sênior	()
Pleno	()	Pleno	()
Júnior	()	Júnior	()
Não conheço	()	Não conheço	()



Pontifícia Universidade Católica do Rio Grande do Sul – PUCRS
Escola Politécnica
Programa de Pós-Graduação em Ciências da Computação

► **Tarefa**

3. Com base na documentação fornecida indique qual(is) tarefa(s) você avaliou e dentre estas qual(is) você selecionaria para executá-la. *

ID	Tarefa	Tecnologia Base	Lido	Selecionado
TJ1	HTML Table data extraction - Ideation challenge	Java	()	()
TJ2	Social Bike Owner App iOS - Backend Services Part 1		()	()
TJ3	XPrize Visioneers Native Android App Cache Implementation & Bug Fixes		()	()
TJ4	QED - Learning Library Mobile Application Ionic Prototype		()	()
TJ5	Panel Schedule Importer - Image De-Noise & In-Paint Machine Learning		()	()
TJ6	SM engine - CRUD Services Integration		()	()
TJ7	SM engine - Proxy Server Integration		()	()
TJ8	TC Review Microservice Local Deployment		()	()
TJ9	TC Identity Service - Groups API Return All Sub-groups		()	()
TA1	Holmes Negative News Web App Integration	Angular	()	()
TA2	[Payment Increased Again] Applet 2 HTML5 - Relationship Manager Bug Fixes		()	()
TA3	Predix - Integration with Alexa Voice Service		()	()
TA4	IBM Cognitive ChatBot - Misc Updates II		()	()
TA5	IBM Cognitive ChatBot - Misc Updates		()	()
TA6	Authentication Solution Angular4 and Material UI Prototype Challenge		()	()



Pontifícia Universidade Católica do Rio Grande do Sul – PUCRS
Escola Politécnica
Programa de Pós-Graduação em Ciências da Computação

Considerando a(s) tarefa(s) que você selecionou acima, responda às questões que seguem.

4. O que lhe fez selecionar a(s) tarefa(s) indicada(s) anteriormente. *

5. O que fez você desconsiderar outra(s) tarefa(s) anteriormente lida(s) e não selecionada(s). *

Avalie as questões que seguem considerando toda a documentação fornecida para a(s) tarefa(s) que você selecionou. Por documentação da tarefa se entende:

- **a descrição da tarefa;** e/ou
- **a documentação complementar** (especificações de regras de negócio ou técnicas, padrões de desenvolvimento, diagramas, critérios de aceitação da tarefa, etc).

6. De modo geral como você classifica a documentação da(s) tarefa(s) que você selecionou para executar? *

Excelente	Muito boa	boa	Fraca	Pobre
()	()	()	()	()



Pontifícia Universidade Católica do Rio Grande do Sul – PUCRS
Escola Politécnica
Programa de Pós-Graduação em Ciências da Computação

7. Como você classifica a documentação da(s) tarefa(s) que você selecionou? Opine considerando cada um dos critérios a seguir e como ela se relaciona com a documentação fornecida. *

ID	Critério de Avaliação	Classificação
C1	A documentação fornecida para a(s) sua(s) tarefa(s) estava(m) devidamente autocontida(s) , ou seja, não necessita de outras informações para executar a tarefa.	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo parcialmente <input type="checkbox"/> Não concordo nem discordo <input type="checkbox"/> Discordo parcialmente <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não se aplica
C2	A documentação fornecida para a(s) sua(s) tarefa(s) possui o nível de detalhamento necessário para executar a(s) tarefa(s), ou seja, descreve as etapas que devem ser realizadas de forma satisfatória para a finalização da tarefa.	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo parcialmente <input type="checkbox"/> Não concordo nem discordo <input type="checkbox"/> Discordo parcialmente <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não se aplica
C3	A documentação fornecida para a(s) sua(s) tarefa(s) possui o nível de precisão necessária para executar a(s) tarefa(s), ou seja, descreve os detalhes de como os procedimentos devem ser realizados para finalizar a tarefa.	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo parcialmente <input type="checkbox"/> Não concordo nem discordo <input type="checkbox"/> Discordo parcialmente <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não se aplica
C4	A documentação fornecida para a(s) sua(s) tarefa(s) está corretamente documentada e coesa de forma a dar o suporte necessário para executar a tarefa.	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo parcialmente <input type="checkbox"/> Não concordo nem discordo <input type="checkbox"/> Discordo parcialmente <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não se aplica
C5	A documentação fornecida para a(s) sua(s) tarefa(s) descreve uma atividade factível dentro do prazo e orçamento proposto .	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo parcialmente <input type="checkbox"/> Não concordo nem discordo <input type="checkbox"/> Discordo parcialmente <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não se aplica
C6	A documentação fornecida para a(s) sua(s) tarefa(s) não possui informações desnecessárias para a execução da(s) tarefa(s).	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo parcialmente <input type="checkbox"/> Não concordo nem discordo <input type="checkbox"/> Discordo parcialmente <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não se aplica
C7	A documentação fornecida para a(s) sua(s) tarefa(s) possui informações dúbias , ou seja, pode apresentar interpretações divergentes conforme o interlocutor que estiver lendo esta documentação.	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo parcialmente <input type="checkbox"/> Não concordo nem discordo <input type="checkbox"/> Discordo parcialmente <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não se aplica
C8	A documentação fornecida para a(s) sua(s) tarefa(s) possui informações suficientes para realizar os testes necessários a fim de validar se a tarefa foi cumprida conforme esperado.	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo parcialmente <input type="checkbox"/> Não concordo nem discordo <input type="checkbox"/> Discordo parcialmente <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não se aplica
C9	A documentação fornecida para a(s) sua(s) tarefa(s) possui problemas que podem gerar defeitos no resultado final da(s) tarefa(s).	<input type="checkbox"/> Concordo totalmente <input type="checkbox"/> Concordo parcialmente <input type="checkbox"/> Não concordo nem discordo <input type="checkbox"/> Discordo parcialmente <input type="checkbox"/> Discordo totalmente <input type="checkbox"/> Não se aplica



Pontifícia Universidade Católica do Rio Grande do Sul – PUCRS
Escola Politécnica
Programa de Pós-Graduação em Ciências da Computação

8. Como base nas respostas fornecidas na questão anterior. justifique e identifique na documentação o que fez você optar pelas suas respostas. *

(você pode destacar suas observações diretamente na cópia da documentação fornecida e se necessário no espaço abaixo. Por exemplo:

No documento TA3 considero o texto muito bom, texto claro sem dúvidas para realizar o solicitado, ou

No documento TJ2 o item C3 tem pouca informação que viabilize criar o logotipo solicitado.)



Pontifícia Universidade Católica do Rio Grande do Sul – PUCRS
Escola Politécnica
Programa de Pós-Graduação em Ciências da Computação

9. Em sua opinião, quais pontos podem ser melhorados ou mantidos em relação a descrição da documentação (enunciado da tarefa e documentação complementar) associada à(s) tarefa(s) que você selecionou de forma a contribuir com a execução da mesma. *



Pontifícia Universidade Católica do Rio Grande do Sul – PUCRS
Escola Politécnica
Programa de Pós-Graduação em Ciências da Computação

10. A partir deste seu contato com a plataforma TopCoder, como você classifica sua motivação para participar de tarefas no modelo de *Software Crowdsourcing*? *

Nenhuma motivação	Pouca motivação	Nem muita nem pouca motivação	Boa motivação	Muita motivação
()	()	()	()	()

11. Justifique sua resposta anterior indicando quais os fatores que lhe motivam (ou não) a participar da plataforma TopCoder. *

TOPCODER

TAREFAS: JAVA

HTML Table data extraction - Ideation challenge

Code TC018 Java Python Other URL: <https://www.topcoder.com/challenge-details/30060070?type=develop>

PRIZES		
1st	2nd	3rd
\$900	\$500	\$300

The challenge is finished. [Show Deadlines](#)

Entrega: 5 dias

[DETAILS](#) [REGISTRANTS \(71\)](#) [SUBMISSIONS \(11\)](#) [CHALLENGE FORUM](#)

Challenge Overview

Unstructured text mining is the key the future of data science. Many data sources provide valuable data in unstructured and semi-structured formats. Frequently, tables in HTML are messy, and complex to find within large bodies of text. Headers change, can be misaligned, numbers can be in different formats. The tables titles are frequently labeled outside the table structure. Mastering the extraction of this information in a generic way can provide a very powerful tool for understanding many different forms of data.

Our goal in this challenge is to get some ideas and a design for a tool that takes a html file as input and as the output produces structured text in a JSON format with data from the tables in the document. Specifically, the tool should be able to extract every number in the table including understanding missing values labeled by:

```
The row name including sub-row names aligned to the tree
The column header including any information for multiple headers
The units for the number e.g. dollars, shares
The date associated with the number from the header eg 2001
Any ancillary information such as if it is a pro forma number or not
The value in absolute units eg 1 million should be 1000000
No commas or other separators in values
The name of the table from any headers above the table or outside table
The date of the document from the index
The company name / ID associated with the table
The originating document type if there is a type
All HTML tags must be stripped from the output
```

As a sample document we can use the SEC Apple 10-Q (link is <https://tinyurl.com/y95t497g>), table on page 3. The expected output for the table data is attached in the forums. you can use other 10-Q filings as sample documents - [Facebook](#), [Alphabet](#), [Microsoft](#), but bear in mind that the tool should be a generic one and not rely on specific formatting data of those sample documents.

This tool is intended to be a component of an architecture. Implementation must be in python, java, or scala. Spark is acceptable as a platform but not required. Any libraries used must be Apache, BSD, or MIT licenced. No GPL is permitted.

Final Submission Guidelines

Submit a document describing your solution. Any additional design documents such as uml are highly desirable. Your proposal will be used as a basis for implementation so make sure it contains all the necessary implementation details.

ELIGIBLE EVENTS:

[2018 TopCoder\(R\) Open](#)

REVIEW STYLE:

Final Review:
Community Review Board 7

Approval:
User Sign-Off 7

[Supa](#)

CHALLENGE LINKS:

[Review Scorecard](#)

CHALLENGE TERMS:

[Standard Terms for TopCoder Competitions v2.1](#)

GET THE UML TOOL:

- [Github source code repository](#)
- [Mac disk image](#)
- [Java installer](#)

SHARE:



TJ1

Payments

Topcoder will compensate members in accordance with our standard payment policies, unless otherwise specified in this challenge. For information on payment policies, setting up your profile to receive payments, and general payment questions, please refer to <https://help.topcoder.com/hc/en-us/articles/217482038-Payment-Policies-and-Instructions>.

Reliability Rating and Bonus

For challenges that have a reliability bonus, the bonus depends on the reliability rating at the moment of registration for that project. A participant with no previous projects is considered to have no reliability rating, and therefore gets no bonus. Reliability bonus does not apply to Digital Run winnings. Since reliability rating is based on the past 15 projects, it can only have 15 discrete values. [Read more.](#)

TJ2

Social Bike Owner App iOS - Backend Services Part 1

Code TCO18 Java PostgreSQL REST Linux URL: <https://www.topcoder.com/challenge-details/30059564/?type=develop>

PRIZES

Entrega: 6 dias

1st: \$1,200

2nd: \$600

The challenge is finished. [Show Deadlines](#)

[DETAILS](#) [REGISTRANTS \(38\)](#) [SUBMISSIONS \(9\)](#) [CHALLENGE FORUM](#)

Challenge Overview

Project Overview

This app is like Pinterest or Tinder for all motorcycle owners. Motorcycle owners post pictures of their bike(s) and identify customizations they have made. Other motorcycle owners can search for and view bikes with the customizations they are interested in. Gamification is added to gather more data and to give it more stickiness.

Competition Task Overview

For this challenge and the next few ones, we need to build the services to support all the features in the iOS app.

For this particular challenge, the following sections of endpoints defined in the swagger documentation are in scope:

1. Security
2. User
3. Motorcycle
4. LikeOrDislike
5. Comment
6. Lookup

General Guidelines

- The services will be hosted on [Rackspace](#), please make sure there's no compability issue.
- ◆◆◆ Code must be clearly documented
- Code must follow the general layers as defined in the architecture
- Provide a separate Swagger API document that covers the APIs implemented in this challenge
- Unit tests are not required, but please provide a postman file with positive and negative inputs

Technology Overview

Java 8
Spring Boot
PostgreSQL
REST
Jetty / Tomcat / JBoss
Maven
Rackspace

Documentation Provided

Register to see documents in the forum.

ELIGIBLE EVENTS:
[2018 TopCoder\(R\) Open](#)

REVIEW STYLE:

Final Review:
[Community Review Board](#)

Approval:
[User Sign-Off](#)

CHALLENGE LINKS:
[Review Scorecard](#)

CHALLENGE TERMS:
[Standard Terms for TopCoder Competitions v2.1](#)
[Appirio NDA v2.0](#)

GET THE UML TOOL:
- [Github source code repository](#)
- [Mac disk image](#)
- [Java installer](#)

SHARE:



TJ2

Final Submission Guidelines

Submission Deliverables

- A single zip file including all files that cover all the mentioned requirements
- A complete and detailed deployment document in markdown format explaining how to deploy the application including configuration information.
- Maven build script to build / test / pack the code.

Final Submission

For each member, the final submission should be uploaded via the challenge detail page on topcoder.com.

Payments

Topcoder will compensate members in accordance with our standard payment policies, unless otherwise specified in this challenge. For information on payment policies, setting up your profile to receive payments, and general payment questions, please refer to <https://help.topcoder.com/hc/en-us/articles/217482038-Payment-Policies-and-Instructions>.

Reliability Rating and Bonus

For challenges that have a reliability bonus, the bonus depends on the reliability rating at the moment of registration for that project. A participant with no previous projects is considered to have no reliability rating, and therefore gets no bonus. Reliability bonus does not apply to Digital Run winnings. Since reliability rating is based on the past 15 projects, it can only have 15 discrete values. [Read more](#).

TJ3

XPrize Visioneers Native Android App Cache Implementation & Bug Fixes

Code Android Java URL: <https://www.topcoder.com/challenge-details/30058995/?type=develop>

PRIZES

Entrega: 5 dias

1st \$1,200

2nd \$600

The challenge is finished. [Show Deadlines](#)

[DETAILS](#) [REGISTRANTS \(36\)](#) [SUBMISSIONS \(2\)](#) [CHALLENGE FORUM](#)

Challenge Overview

NOTE - This challenge has shorter phases and a fixed reviewer fee.

Welcome to the XPrize Visioneers Native Android App Resources Implementation & Bug Fixes challenge.

In the earlier challenges, we've built the various screens for this app and then hooked it up to the client API. The code is available on our Github repository (request access via forums). Now, we want to implement the pending screen for Resources and fix some issues found in client feedback.

Challenge Requirements

The scope of this challenge involves

A) Client Feedback bugs

We've provided a slide deck in the forums which lists out the app issues based on client review

B) Implement App Caching

We need to implement caching behavior in app as per the following rules

1. The user must be able to login offline and all data must be retrieved from the app cache
2. Create the following values in app settings. Set default cache values to the following (in minutes)

- "Cache Expiry (minutes)" - 1 Day
- "Interval for Checking New Messages" and set that default to 30 mins
- "Reset Cache on Next Launch"

3. Caching Rules:

- Rule for all screens should be the same. The data received from server must be cached until the 'Cache Expiry' parameter is hit. Caching should not prevent new data from coming in (i.e. messages)... Caching should only apply to existing data.
- Setting 'Reset Cache on Next Launch' - should clear the cache for all data for tabs for the next app

REVIEW STYLE:

Final Review:
Community Review Board 

Approval:
User Sign-Off 

CHALLENGE LINKS: [Review Scorecard](#)  [Support](#)

CHALLENGE TERMS:
[Standard Terms for TopCoder Competitions v2.1](#)

GET THE UML TOOL:
- [Github source code repository](#)
- [Mac disk image](#)
- [Java installer](#)

SHARE:



TJ3

- Setting 'Reset Cache on Next Launch' - should clear the cache for all data for tabs for the next app launch

C) Other Fixes

- 1) Need to add option to **"Refresh Data"** on the User Profile Page. This should be the first item, above Terms & Conditions. This should invoke the corresponding API again.
- 2) List the app version number under settings. Use v0.1 for this version
- 3) Integrate the provided app icons
- 4) Background image for the prizes should change when a new prize is selected and still appears to be hard-coded. They should be using the `api/sponsor/{id}/prizes` field `prizeBannerImageUrl` for the background image.
- 5) Need to add privacy policy on the login screen (attached in forums)
- 6) When going offline, we are still getting a "No Internet Connection Screen". This screen should not appear. The cached data should be displayed and if no cache data is available then display "No Internet Connection Available".

General Notes

Note the following :

- The app remains locked to portrait mode
- We're open to using Retrofit, Volley and okhttp. If you need to use another library/ framework, please ask for approval in the forums to ensure there are no license violations
- Make the Server URL configurable, do NOT hardcode any parameters directly in your code
- Make sure all your code is well-documented using Javadoc comments. All public methods and attributes must have valid Javadoc comments.

Final Submission Guidelines

- Android Studio project
- Deployment guide with configuration & verification steps. Describe all config params and include steps to run
- Demo video (Mandatory and will be scored down if not provided)
- Winner will be asked to raise a merge request on the repo

Payments

Topcoder will compensate members in accordance with the our standard payment policies, unless otherwise specified in this challenge. For information on payment policies, setting up your profile to receive payments, and general payment questions, please refer to <https://help.topcoder.com/hc/en-us/articles/217482038-Payment-Policies-and-Instructions>.

TJ3

Reliability Rating and Bonus

For challenges that have a reliability bonus, the bonus depends on the reliability rating at the moment of registration for that project. A participant with no previous projects is considered to have no reliability rating, and therefore gets no bonus. Reliability bonus does not apply to Digital Run winnings. Since reliability rating is based on the past 15 projects, it can only have 15 discrete values. [Read more](#)

TJ4

QED - Learning Library Mobile Application Ionic Prototype

Code Android HTML5 Ionic JavaScript iOS Mobile URL: <https://www.topcoder.com/challenge-details/30058907/?type=develop>

PRIZES

	1st	2nd
Entrega: 6 dias	\$1,400	\$700

The Challenge is finished. [Show Deadlines](#)

[DETAILS](#) [REGISTRANTS \(10\)](#) [SUBMISSIONS \(2\)](#) [CHALLENGE FORUM](#)

Challenge Overview

Challenge Overview

QED is a mobile application that will be used to help users find information that at the same time will help them solve some problems. We have completed the design for both the desktop web dashboard and the mobile app. For this challenge, we'd like to convert the design to an ionic based prototype app that will run on iOS and Android devices.

Please only focus on the mobile design for this challenge.

All screens from the mobile design are in scope.

Assets provided in Forums (available on registration)

MarvelApp Link

Design Sources

Screens Requirements

Overall

- Overall the screen flow is pretty clear if you follow the MarvelApp link, so please carefully check that.
- Show hover/active states for buttons, dropdowns, breadcrumbs, errors/success states, elements with interaction, etc.

Screen Notes

- 01-00Welcome-Screens.jpg: for this screen, just play some video from Youtube for now.
- 04-3Notifications.jpg: when View Old Notifications is pressed, just load more notifications into the list and make it scrollable (if not already). Push notification is NOT in scope of this prototype.
- 04-4Search-Content.jpg: remove the Content and Tags tabs, just keep the search bar and the recent searches sections.
- 05-1Content-Details.jpg: for video type content, hitting the image should bring up a player that plays a video from Youtube; for audio type content, hitting the image should play the audio.
- 08-1Settings.jpg: ignore the Update Password section here.

REVIEW STYLE:

Final Review:
Community Review Board ?

Approval:
User Sign-Off ?

CHALLENGE LINKS:

[Review Scorecard](#)

CHALLENGE TERMS:

[Standard Terms for TopCoder Competitions v2.1](#)
[Apprio NDA v2.0](#)

GET THE UML TOOL:

- [Github source code repository](#)
- [Mac disk image](#)
- [Java installer](#)

SHARE:



TJ4

General Notes

- You are building an Ionic Cordova app in this contest, with Android and iOS as the target platforms. Please use Ionic 2.
- The app will be locked to Portrait mode. Landscape mode is NOT in scope.
- Consider iOS 10.0+ and Android 5.0 as the baseline versions.
- The application is for phones and is expected to run on tablets in scaled mode.
- The design asset sources and Marvelapp link are provided in the forum.
- Use Ionic components/directives (ion-list, ion-item, etc.) as much as possible, providing they exactly match the behaviour and styling of the HTML prototype (see below). You can use the Sass variables for the Ionic directives for customising the style to fit the app.
- If the Ionic component/directive does not exactly match the design, then you should create an Ionic component/directive.
- Consider how you might employ view caching to improve the user experience.
- We strongly recommend using [this seed project](#) as a starting point. Clean up anything that you don't need.
- The converted prototype should exactly match the provided design including the screen transitions.
- For data for all screens, use JSON files and use methods that read data off these files. The idea is that we should be able to replace the URL in these methods to point to actual backend at a later stage for minimal changes. Do NOT hardcode data into the code.

Final Submission Guidelines

Submission Deliverables

- Zipped project source code
- Verification video covering both iOS & Android
- Provide detailed deployment document that covers the following:
 - How to setup the environment, required software and dependencies in order to run the apps
 - How to build the app, package it into both iOS and Android platforms, and run it in emulators

Payments

Topcoder will compensate members in accordance with our standard payment policies, unless otherwise specified in this challenge. For information on payment policies, setting up your profile to receive payments, and general payment questions, please refer to <https://help.topcoder.com/hc/en-us/articles/217482038-Payment-Policies-and-Instructions>.

Reliability Rating and Bonus

For challenges that have a reliability bonus, the bonus depends on the reliability rating at the moment of registration for that project. A participant with no previous projects is considered to have no reliability rating, and therefore gets no bonus. Reliability bonus does not apply to Digital Run winnings. Since reliability rating is based on the past 15 projects, it can only have 15 discrete values. [Read more.](#)

TJ5

Panel Schedule Importer - Image De-Noise & In-Paint Machine Learning

Code **TCO18** C# Cognitive Docker Java Other Python Linux Microsoft Azure URL: <https://www.topcoder.com/challenge-details/30058878?type=develop>

PRIZES

Entrega: 10 dias	1st \$1,200	2nd \$600
------------------	-----------------------	---------------------

The challenge is finished. [Show Deadlines](#)

[DETAILS](#) [REGISTRANTS \(44\)](#) [SUBMISSIONS \(4\)](#) [CHALLENGE FORUM](#)

Challenge Overview

Welcome to Panel Schedule Importer – Image De-Noise & In-Paint challenge. This contest is part of the Panel Schedule Importer series where our client attempts to extract data from PDFs and images using OCR and Machine Learning. The goal of this challenge is to remove one or more types of annotation artifacts that obstruct the OCR process. We refer to these artifacts as "noise"

This contest has bonus payments associated with it. Please read the specification below.

Project Overview

Electrical engineers send our client, a Fortune Global 500 company specializing in energy management, technical documents (in PDF format) that include one or more electrical panel configuration(s). Configuration data within these documents describes the type and amp rating of circuit breakers, a description of the equipment served by a breaker, and general specifications for the panel itself. These sections of the document are called schedules and are depicted in table format.

Importing data from the schedule tables is difficult for a few reasons. For one, various engineering firms use different table formats to articulate this information. Additionally, there is significant variation in the quality and/or completeness of table images.

This project seeks to automate the task of identifying the schedules in the PDFs and extracting meaningful data from it.

Contest Details

This is the third challenge in the Panel Schedule Importer series where we are attempting to reduce the amount of human intervention required to extract panel specifications from technical documents to prepare customer quotes.

Panel schedules within a PDF are sometimes delivered in a 'clean' format that is ready for immediate OCR processing. In some cases, however, schedules are annotated with superimposed notes (usually digital graphics created with PDF editing software – but may also be handwritten notes) that obscure important text and interfere with OCR.

ELIGIBLE EVENTS:

2018 TopCoder(R) Open

REVIEW STYLE:

Final Review:
Community Review Board ?

Approval:
User Sign-Off ?

[Supp](#)

CHALLENGE LINKS:

[Review Scorecard](#)

CHALLENGE TERMS:

[Standard Terms for TopCoder Competitions v2.1](#)
[Apprio NDA v2.0](#)

GET THE UML TOOL:

- [Github source code repository](#)
- [Mac disk image](#)
- [Java installer](#)

SHARE:



TJ5

For this challenge, we will provide you with 36 sample images in BMP format. Each image contains one or more schedule tables. Your task is to provide a solution for improving image quality and preparing images for OCR data capture. Images must be processed to remove (or reduce) background noise in order to improve the accuracy of OCR.

There are several common types of noise that need to be removed, including:

1. bubble-shaped graphic overlays
2. geometric shape graphic overlays (which typically have a number at the center)
3. background noise/ poor contrast as the result of a poor image scan
4. handwritten text and/or shapes

Considering the process of removing these noise artifacts as applying filters, you can thus apply 4 filters to the images provided, that remove the different types of noise. For this contest, you are required to implement one of the first two mentioned above (bubble shaped and geometric shape). Implementing just one filter is fine. However, implementing both will result in a bonus payment of \$600. These will be reviewed under Major Requirements in the scorecard. Additionally, the third and fourth filters are optional (background noise and handwritten text) and if implemented, they will be reviewed under Minor Requirements in the scorecard. Also, each attract a bonus payment of \$150. Thus, if your submission receives first or second place, you are eligible to get the bonus in addition to receiving payment associated with the first or second place.

This implies that not all 36 images that we share will be relevant for your submission. Only those images that correspond to the filter category implemented by your submission will be used during review.

Additionally, these overlays and other forms of noise sometimes obscure text content or cause it to be unreadable by OCR. Similarly, the process of reducing "noise" may itself cause text content to be lost. In these situations, an algorithm/ Machine Learning approach should be used to automatically detect for missing content and attempt to "fill-in" any missing data. This is optional. If carried out, this attract an additional payment of \$400 if the submission receives first or second place. This is in addition to receiving payment associated with the first or second place.

You will find examples of the types of noise in a document shared in the contest forum.

The goal of this challenge is to provide an image pre-processing application to aid in downstream OCR processing. The following article details a similar application, written in Python, for use with Tesseract OCR – and may serve as a resource for competitors: <http://www.pyimagesearch.com/2017/07/10/using-tesseract-ocr-python/>

For this contest, a successful submission should perform the following functions:

1. Accept a rasterized image (.BMP format) as input
2. Test/evaluate image quality to determine if annotations, overlays or other background noise are contained in the image and which interfere with legibility of text
 1. If images does not contain background noise, proceed with next stages of processing

TJ5

1. If image does not contain background noise, proceed with next stage of processing
2. If image contains background noise, remove or reduce all noise that would interfere with/ reduce the accuracy of OCR & proceed with next stage of processing
3. Test/evaluate the resulting "de-noised" image for missing or obfuscated text
 1. If image is 'complete' (does not contain missing content or obfuscated text), flag the file as 'ready' for OCR processing
 2. If critical text content is missing from the image, use surrounding context to populate missing data with appropriate content using a "best match" approach
4. Output the fully processed image in BMP format, preserving the native image dimensions and resolution
5. Output a CSV log file that contains details on the image processing. The minimum columns should include Original Filename and Output Filename, as well as indicators whether the image contained noise and whether text was missing/ required populating. Optionally, your log file could include data quantifying the "amount of noise" detected – and/or a confidence score for re-populating text. An ideal solution might also contain additional details like image dimensions and file size.

Points to Note

- Since your solution will likely involve complex dependencies and environmental setups, it is desired that your solution includes a Dockerfile. This is not required however it should eliminate environmental discrepancies that may prevent judges from reviewing your submission.
- You are only allowed to make use of MIT licensed, BSD licensed, Mozilla licensed or Apache licensed libraries in your solutions.
- Optional: You may wrap your solution in a script that will cycle through the entire set of 36 BMP images or you may allow for it to be called one time for each BMP
- **Note that you can submit your solution in Python, Java or C#**
- The major requirements of this contest are:
 - Implement the bubble shapes and / or geometric shape removal filters. In implementing the filters, you are expected to reduce or remove the noise that would reduce the accuracy of OCR processing of the image in future stages.
 - Output the processed image in BMP format, preserving the native image resolution and dimensions
 - Output a CSV log file that contains details on the image processing. The details of the columns in the log file have been described above.
 - If implemented, the optional fill-in feature - that of filling in missing content after reducing noise will also be a major requirement. If not implemented, considering it to be optional, reviewers will not deduct scores.

Final Submission Guidelines

- Include a detailed deployment guide (a README file is also fine as long as it contains deployment instructions) along with your source code and upload it to Topcoder.
- Don't forget to include an unlisted link to your video that shows your solution in action.
- Do mention which among the four filters did you provide

TJ5

- Do mention which among the four filters did you provide.

Payments

Topcoder will compensate members in accordance with the our standard payment policies, unless otherwise specified in this challenge. For information on payment policies, setting up your profile to receive payments, and general payment questions, please refer to <https://help.topcoder.com/hc/en-us/articles/217482038-Payment-Policies-and-Instructions>.

Reliability Rating and Bonus

For challenges that have a reliability bonus, the bonus depends on the reliability rating at the moment of registration for that project. A participant with no previous projects is considered to have no reliability rating, and therefore gets no bonus. Reliability bonus does not apply to Digital Run winnings. Since reliability rating is based on the past 15 projects, it can only have 15 discrete values. [Read more](#).

TJ6

SM engine - CRUD Services Integration

Code TCO18 Java Spring Heroku URL: <https://www.topcoder.com/challenge-details/30058856?type=develop>

PRIZES

	1st	2nd
Entrega: 5 dias	\$1.000	\$600

The challenge is finished. [Show Deadlines](#)

[DETAILS](#) [REGISTRANTS \(49\)](#) [SUBMISSIONS \(2\)](#) [CHALLENGE FORUM](#)

Challenge Overview

In this challenge series we'll be developing a SSO solution that will be a reverse proxy to a target application. It will authenticate users via LDAP, then authorise logged in user to perform requested action. Authorization will use a set of rules and policies, and will set additional HTTP headers to the proxied request. Application will have a separate management interface to manage policies, rules, headers and users.

In a [previous challenge](#), we have built crud services for managing the SSO application. In this challenge we'll update api security and add a service to evaluate application policies, rules and headers. Here are the exact requirements:

1. Update authentication to use JWT token (header Authorization: Bearer JWT)
2. Add LoginController with a single endpoint - login(username,password). Actual login will be done by the LdapService. On success create and return a JWT token and user info.
3. Update com.sm.engine/security/* to use the new JWT token
4. Update Header.userAttributes to be a list of DynamicHeader (String headerName, String attributeName) - this includes updating models, services, controllers, swagger and postman.
5. Current crud services will accept fully populated referenced entities in create/update methods, but will not create/update the referenced entities - just integrity checks are performed using entity id. Update the services to actually insert the referenced entities if object ids are null, and keep reference checks when object ids are populated. Make sure to update swagger and postman files and remove any redundant parameters. Right now both swagger and postman api calls contain all the data for referenced entities even though only ids are used.
6. Add /headers/evaluate/{username} endpoint to HeaderController and the same method to HeaderService. It will return a list of objects { "name": "header_name", "value": "header_value" }. Remove the "getHeadersPreview" method from HeaderService.

HeaderService.evaluate will do the following

- Select all enabled policies from the database using PolicyService search. PolicyService.createSearchExample should be updated to support searching for enabled policies (Policy.enabled field)
- For each of the policies evaluate its rules by calling LDAPService.evaluateRule(ruleString), with ruleString being comma joined list of name=value pairs from rule.ruleInfo (for example if ruleInfo contains [{name:"cn",value:"test"},{name:"ou",value:"test1"}], ruleString should be "cn=test,ou=test1"). If all policies evaluate to true, the policy is active and it's headers should be added to the response
- For each header in active policies do the following
 - For each static variable in the header add variable name and value to the result
 - For user attributes call LDAPService.getUserAttributes to get values for user attributes and add all of them to the result

NOTE: For LDAP operations, we have [developed a separate service](#) that you should integrate with the API. LDAPService.evaluateRule will be a necessary method that is included as a dependency in

ELIGIBLE EVENTS:

[2018 TopCoder\(R\) Open](#)

REVIEW STYLE:

Final Review:
Community Review Board 2

Approval:
User Sign-Off 2

[Supa](#)

CHALLENGE LINKS:

[Review Scorecard](#)

CHALLENGE TERMS:

[Standard Terms for TopCoder Competitions v2.1](#)

GET THE UML TOOL:

- [Github source code repository](#)
- [Mac disk image](#)
- [Java installer](#)

SHARE:



TJ6

NOTE: For LDAP operations, we have [developed a separate service](#) that you should integrate with the API project. LDAP service should still be a separate project that is included as a dependency in pom.xml. The provided LDAP_service project is a full spring boot application and you should refactor it to include only the service layer - application and controller classes can be removed. Change the package name to com.sm.engine. Include instructions for local and Heroku deployment. All API changes should be reflected in the swagger and postman files as well.

Final Submission Guidelines

Submit the full source code for the app
Deployment guide (environment, configuration, build, run - local and heroku)
Verification guide for testing the api

Payments

Topcoder will compensate members in accordance with our standard payment policies, unless otherwise specified in this challenge. For information on payment policies, setting up your profile to receive payments, and general payment questions, please refer to <https://help.topcoder.com/hc/en-us/articles/217482038-Payment-Policies-and-Instructions>.

Reliability Rating and Bonus

For challenges that have a reliability bonus, the bonus depends on the reliability rating at the moment of registration for that project. A participant with no previous projects is considered to have no reliability rating, and therefore gets no bonus. Reliability bonus does not apply to Digital Run winnings. Since reliability rating is based on the past 15 projects, it can only have 15 discrete values. [Read more](#).

TJ7

SM engine - Proxy Server Integration

Code Java Spring Heroku URL: <https://www.topcoder.com/challenge-details/30058855/?type=develop>

PRIZES

	1st	2nd
Entrega: 5 dias	\$1,200	\$600

The challenge is finished. [Show Deadlines](#)

[DETAILS](#) [REGISTRANTS \(38\)](#) [SUBMISSIONS \(3\)](#) [CHALLENGE FORUM](#)

Challenge Overview

In this challenge series we'll be developing a SSO solution that will be a reverse proxy to a target application. It will authenticate users via LDAP, then authorize logged in user to perform requested action. Authorization will use a set of rules and policies, and will set additional HTTP headers to the proxied request. Application will have a separate management interface to manage policies, rules, headers and users.

In a [previous challenge](#), we have implemented a sample proxy server using Zuul library that demonstrates how to proxy the requests, add headers to the request, or deny proxying the request by redirecting to login page.

In this challenge we'll build on the sample proxy server and add the features to authorize users and set necessary HTTP headers. The entire flow should go like this:

- When a new request is received, get the 'token' parameter from query string (sample request url is `https://example.net/PWebNG/Modules/PWebEntry/DispatcherView.aspx?token=u8Y3Oq6w5xRxxYwb4JMAjRRw6zKBdvsr`)
- Use the token to call a SOAP service called LandingPageDealer to get user request context. Sample request body can be found in the Functional spec. Url of the service should be configurable
- Response will contain frontEndSystem field that can have two values 'PartnerCenter' or 'BWEB' (it determines if the request is coming from Partner Center of Business Web). In case of BWEB, the request needs to be authenticated before being actually proxied. To authenticate the user, first we need to serialize the original request and save it to user session. Then we will redirect the user to a login page with username and password. Login page should be a part of the proxy server project, not an external page. On form submit, call `LDAPService.login(username,password)`. In case of failed login, show the login page again, otherwise retrieve the original request, remove it from the session and proceed to the next step. For LDAP operations, we have [developed a separate service](#) that you should integrate with the proxy server. LDAP service should still be a separate project that is included as a dependency in pom.xml. The provided `LDAP_service` project is a full spring boot application and you should refactor it to include only the service layer - application and controller classes can be removed. Change the package name to `com.sm.engine`
- The original request needs to be proxied to the target server but first we need to add HTTP headers to the request. To get the list of headers call out backend API (use a mock server for now). The format should be `GET {api_path}/headers/evaluate/{username}`, `Authorization: Bearer {api_token}`. `api_path` and `api_token` should be configurable values, and `username` should be the username of the current user (in case of Partner Center, use a `Login` field from the `LandingPageDealer` response). Response will contain a list of objects (`"name","header_name","value","header_value"`). Add all the returned headers to the request
- Forward the request to the target server and return the response to the client

Include instructions for local and Heroku deployment.

REVIEW STYLE:

Final Review:
Community Review Board 7

Approval:
User Sign-Off 7

CHALLENGE LINKS:

[Review Scorecard](#)

CHALLENGE TERMS:

[Standard Terms for TopCoder Competitions v2.1](#)

GET THE UML TOOL:

- [Github source code repository](#)
- [Mac disk image](#)
- [Java installer](#)

SHARE:



TJ7

Include instructions for local and Heroku deployment.

Final Submission Guidelines

Submit the full source code for the app

Deployment guide (environment, configuration, build, run - local and heroku)

Verification guide for testing the server

Payments

Topcoder will compensate members in accordance with the our standard payment policies, unless otherwise specified in this challenge. For information on payment policies, setting up your profile to receive payments, and general payment questions, please refer to <https://help.topcoder.com/hc/en-us/articles/217482038-Payment-Policies-and-Instructions>.

Reliability Rating and Bonus

For challenges that have a reliability bonus, the bonus depends on the reliability rating at the moment of registration for that project. A participant with no previous projects is considered to have no reliability rating, and therefore gets no bonus. Reliability bonus does not apply to Digital Run winnings. Since reliability rating is based on the past 15 projects, it can only have 15 discrete values.

[Read more](#)

TJ8

TC Review Microservice Local Deployment

Five2Finish TCO18 Docker Java Linux URL: <https://www.topcoder.com/challenge-details/30059622?type=develop>

PRIZES

Entrega: 12 dias
(realizado em 8 dias)

EST

\$200

Unregister

Submit

The challenge is finished. Show Deadlines ▾

DETAILS
REGISTRANTS (31)
SUBMISSIONS (3)
CHALLENGE FORUM

Challenge Overview

For this challenge, we need your help to figure out how to run the tc review microservice locally and update the README with a section just for local deployment so that can share it in upcoming challenges with the community for local development.

Please register to see the code in challenge forum.

Final Submission Guidelines

Submit a new or updated README with detailed steps to configure, run and verify the code locally. Using Docker is fine.

Payments

Topcoder will compensate members in accordance with the our standard payment policies, unless otherwise specified in this challenge. For information on payment policies, setting up your profile to receive payments, and general payment questions, please refer to <https://help.topcoder.com/hc/en-us/articles/217482038-Payment-Policies-and-Instructions>.

Reliability Rating and Bonus

For challenges that have a reliability bonus, the bonus depends on the reliability rating at the moment of registration for that project. A participant with no previous projects is considered to have no reliability rating, and therefore gets no bonus. Reliability bonus does not apply to Digital Run winnings. Since reliability rating is based on the past 15 projects, it can only have 15 discrete values. [Read more](#).

ELIGIBLE EVENTS:
2018 TopCoder(R) Open

REVIEW STYLE:
Final Review:
Community Review Board ?

Approval:
User Sign-Off ? ? Supc

CHALLENGE LINKS:
Review Scorecard

CHALLENGE TERMS:
✔ Standard Terms for TopCoder Competitions v2.1
✔ Appirio NDA v2.0

GET THE UML TOOL:

- Github source code repository
- Mac disk image
- Java installer

SHARE:

TJ9

TC Identity Service - Groups API Return All Sub-groups

First2Finish TC018 Java REST SQL AWS URL: <https://www.topcoder.com/challenge-details/30058974?type=develop>

PRIZES

1st **\$250**

Entrega: 10 dias
(realizado em 6 dias)

Unregister

Submit

The challenge is finished. [Show Deadlines](#)

[DETAILS](#) [REGISTRANTS \(29\)](#) [SUBMISSIONS \(9\)](#) [CHALLENGE FORUM](#)

Challenge Overview

Challenge Requirements

Our security system is designed based on RBAC (Role Based Access Control). Users are directly assigned to some roles for performing permitted works.

In previous challenges, we have created groups management api, in this challenge we need to enhance it a bit:

- Update the endpoint so that we can ask the API to return one level of sub-groups or all sub-groups recursively for a specified parent group.
- Update / add unit tests to work with the changes.
- Update swagger yml to reflect the changes.

The code that you need to update is in the folder `tech core/tech core.service.identity`, you should check the README in the `tech core/tech core.service.identity/docker` to see how to deploy and test the service code locally.

Test

Unit tests are required for the new changes. Existing unit tests must also be updated to make sure they still work.

Code

Your code must be well documented.

Technologies

- Java
- Informix
- SQL
- Docker

Final Submission Guidelines

Submission

- Updated code that covers all the requirements.
- A detailed deployment / verification guide explaining how to run and test your submission with some meaningful data.
- The winner will be asked to send a pull request to our repo.

ELIGIBLE EVENTS:

2018 TopCoder(R) Open

REVIEW STYLE:

Final Review:
Community Review Board ?

Approval:
User Sign-Off ?

CHALLENGE LINKS:

[Review Scorecard](#)

CHALLENGE TERMS:

- ✓ Standard Terms for TopCoder Competitions v2.1
- ✓ Appirio NDA v2.0

GET THE UML TOOL:

- [Github source code repository](#)
- [Mac disk image](#)
- [Java installer](#)

SHARE:



TJ9

Payments

Topcoder will compensate members in accordance with the our standard payment policies, unless otherwise specified in this challenge. For information on payment policies, setting up your profile to receive payments, and general payment questions, please refer to <https://help.topcoder.com/hc/en-us/articles/217482038-Payment-Policies-and-Instructions>.

Reliability Rating and Bonus

For challenges that have a reliability bonus, the bonus depends on the reliability rating at the moment of registration for that project. A participant with no previous projects is considered to have no reliability rating, and therefore gets no bonus. Reliability bonus does not apply to Digital Run winnings. Since reliability rating is based on the past 15 projects, it can only have 15 discrete values. [Read more](#).

TOPCODER

TAREFAS : ANGULAR

TA1

Holmes Negative News Web App Integration

[Code](#)
[TE018](#)
[Angular 2+](#)
[CSS](#)
[HTML5](#)
[JavaScript](#)
[MongoDB](#)
[Node.js](#)
[REST](#)
[Linux](#)

URL: <https://www.topcoder.com/challenge-details/30059731/?type=develop>

PRIZES

1st	2nd
\$800	\$400

Entrega: 5 dias

The challenge is finished [Show Deadlines](#)

[DETAILS](#)
[REGISTRANTS \(34\)](#)
[SUBMISSIONS \(3\)](#)
[CHALLENGE FORUM](#)

Challenge Overview

Project Overview

The goal of this project is to create a web application that can do the following:

- Extraction of Subsidiaries Information from pre-defined sources for given entity
- Configure the web sources and search the negative news for Entity Name, Entity Controllers, Owner's name and Subsidiaries name
- Present the consolidated negative news report

For this challenge we need your help to integrate the Angular based prototype with the backend API services so it becomes a fully functional web app.

CHALLENGE REQUIREMENTS

- The web app must be using the backend services to perform login, manage search parameters (admin only), perform search (all logged in users), view results, filter results and download news as PDF, etc.
- You must make sure the web app no longer uses any local data or hardcoded data, everything should be done through the services.
- You should provide a complete deployment guide (in markdown format) that covers how to deploy and configure the various services used, and then how the frontend (web app) should be configured and deployed to work with them. The guide should also include some verification details.
- Right now the search results show in page page, but it should support paging which needs to be taken care of in this challenge.
- You may have to update some of the backend services in order to make the features fully work.
- Must follow Angular best coding practices
- Please stick with the prototype for UI look & feel, and browser compatibility.

TECHNOLOGIES

Javascript
 HTML5
 CSS
 Angular 2+
 Node.js
 REST
 MongoDB
 Microservice

Final Submission Guidelines

Submit a single zip file that includes all the following:

- Updated (iterated) web app code, including detailed readme in markdown format covering how to configure, deploy and verify the app.
- Include any updated backend services in their own folder if you update any of them in this challenge.

Payments

Topcoder will compensate members in accordance with the our standard payment policies, unless otherwise specified in this challenge. For information on payment policies, setting up your profile to receive payments, and general payment questions, please refer to <https://help.topcoder.com/hc/en-us/articles/217482038-Payment-Policies-and-instructions>.

ELIGIBLE EVENTS:

2018 TopCoder(TC) Open

REVIEW STYLE:

Final Review
Community Review Board ?

Approval:
User Sign-Off ?

CHALLENGE LINKS:

Review Scorecard

CHALLENGE TERMS:

Standard Terms for TopCoder Competitions v2.1
Apprio NDA v2.0

GET THE UML TOOL:

- Github source code repository
- Mac disk image
- Java installer

SHARE:



TA1

Reliability Rating and Bonus

For challenges that have a reliability bonus, the bonus depends on the reliability rating at the moment of registration for that project. A participant with no previous projects is considered to have no reliability rating, and therefore gets no bonus. Reliability bonus does not apply to Digital Run winnings. Since reliability rating is based on the past 15 projects, it can only have 15 discrete values. [Read more.](#)

TA2

[Payment Increased Again] Applet 2 HTML5 - Relationship Manager Bug Fixes

Functionality 100% AngularJS 2+ NodeJS URL: <https://www.topcoder.com/challenge-details/30060071/?type=develop>

Prize: **Entrega: 10 dias**
(realizado em 7 dias)

Prize: \$600

The challenge is finished. [Show Deadlines](#)

[Unregister](#) [Submit](#)

[DETAILS](#) [REGISTRANTS \(36\)](#) [SUBMISSIONS \(6\)](#) [WINNERS \(1\)](#) [CHALLENGE FORUM](#)

Challenge Overview

Welcome to Applet 2 HTML5 - Relationship Manager Bug Fixes Contest. We will be carrying out some minor fixes as part of this contest for an existing Angularjs app.

Payment has been increased for this contest.

Contest Details

We are trying to replicate an existing applet as a web application using Angularjs (v4). We have some minor fixes we'd like you to carry out as part of this contest. The fixes are as follows. Before checking them out, launch the app, click on Search and then click any existing result's "To Editor" column twice. This will take you to the editor.

- The editor will have two modes - Selection mode and Create Relation mode. Selection mode is active by default and can be activated manually by clicking on Selection Tool on the left sidebar. To go to Create Relation mode, click on the Create Relation Tool icon in the left sidebar.
 - In Create Relation mode, currently, you have to Shift + Click an entity which then creates a relationship. You can drag and drop this relationship onto another entity to create a relationship.
 - Change this existing behaviour. Let the user click and hold on the entity (and NOT Shift + Click) and then drag the mouse pointer and drop it onto another entity - during the drag, the relationship should show up as it does currently on Shift + Click and on drop it should attach itself to another entity. When this action completes, immediately switch over to the Selection mode and the newly created relationship should be shown.
 - There is a possibility that the user can click an entity and then drop it on an empty space on the editor. When this happens, immediately switch to Selection mode and the earlier created relationship disappears (since it has no destination entity to attach to).
 - When creating a relationship, if the relationship has a destination entity to attach to, after switching over to the Selection mode, show the Create Relationship screen in the right sidebar (image will be shared in the contest forum). No further actions need to be supported on this screen other than showing it.
 - Currently, when creating a relationship, an API call is made. Remove this. No API call is to be made when creating a relationship.
 - Finally, in Create Relationship mode, disable context menus - they are currently shown for Entities, Relationships and on the editor itself (right click on an empty space).
- Selecting entities. In selection mode, you can do this currently by double clicking the entity. Change this behaviour. Let a single click select the entity - change its border to blue and show four small circles at the corners of the entity. You DONT have to show the entity details screen in the right. Let this behaviour exist for Relations as well - in Selection mode, single click on relationship should select it - change its color to blue.
- Selecting multiple entities. You can select multiple entities by using the mouse to create an area of selection (just like selecting multiple folders in Windows). Provide another way to do this - let the user Shift + Click an entity to select multiple entities at the same time. The user can also select a relationship in this process along with the entity by Shift + Clicking it.
- Currently, selecting an entity also selects its outgoing relationships. Correct this so that selecting an entity only select the entity and not its relationships.

ELIGIBLE EVENTS:

2018 TopCoder(H) Open

REVIEW STYLE

Final Review:
Community Review Board 7

Approval:
User Sign-Off 2

CHALLENGE LINKS:

Review Scorecard

CHALLENGE TERMS:

- Standard Terms for TopCoder Competitions v2.1
- Applet NDA v2.0

GET THE UML TOOL:

- GitHub source code repository
- Mac disk image
- Java installer

SHARE:



TA2

Final Submission Guidelines

Carry out the fixes and upload your code base to Topcoder.

Payments

Topcoder will compensate members in accordance with our standard payment policies, unless otherwise specified in this challenge. For information on payment policies, setting up your profile to receive payments, and general payment questions, please refer to <https://help.topcoder.com/hc/en-us/articles/217482038-Payment-Policies-and-Instructions>.

Reliability Rating and Bonus

For challenges that have a reliability bonus, the bonus depends on the reliability rating at the moment of registration for that project. A participant with no previous projects is considered to have no reliability rating, and therefore gets no bonus. Reliability bonus does not apply to Digital Run winnings. Since reliability rating is based on the past 15 projects, it can only have 15 discrete values. [Read more.](#)

TA3

Predix - Integration with Alexa Voice Service

Code: Angular 2+ | AngularJS (1.x) | Express | Node.js | Predix | ReactJS | NodeJS | URL: <https://www.topcoder.com/challenge-details/30058889/?type=develop>

PRIZES

1st	2nd	3rd	4th
\$1,200	\$600	\$200	\$100

The challenge is finished. [Show Deadlines](#)

DETAILS | REGISTRANTS (47) | SUBMISSIONS (2) | CHALLENGE FORUM

Challenge Overview

Our client, General Electric (GE) provides **Predix** - The platform for Industrial Internet of Things (IIoT), which helps to develop, deploy, and operate industrial apps at the edge and in the cloud. It securely connects machines, data, and analytics to improve operational efficiency.

GE wants the Topcoder community to work with Predix and to create sample applications that emulate various devices interoperating with Predix services over the network. This challenge is a part of a series of Predix contests with the total prize pool over \$10,000. We hope you will take time to register for Predix, read about the services they provide, and configure your environment. Other challenges in this series may use different technology stacks and approach very different problems, but all of them will utilize Predix account and different services provided by the platform.

Setup

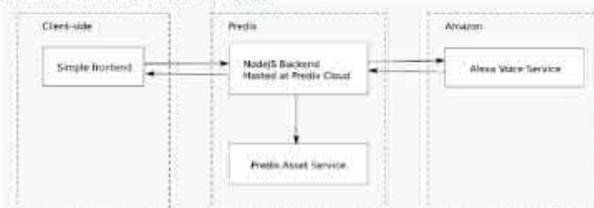
If you have not registered already, go to <http://predix.topcoder.com> and join Topcoder Predix Community. Fill out the registration form and confirm your email address. Once completed, you will receive community welcome email with a link to register your Predix developer account.

During registration of the Predix dev account, state Topcoder as your company. Be aware that approval of the Predix account can take a few days at the Predix side. To facilitate the process, please message in the challenge forum your Topcoder Handle / First and Last Name / Email that you have used for registration of Predix account. We will pass these data to GE. Predix team to speed-up the process.

Eligibility: As of now, residents of the following countries are not allowed to register in Predix, hence they won't be able to participate in this challenge: China, Cuba, Iran, North Korea, Russia, Sudan, Syria.

Challenge Scope

In this challenge we will build a sample webapp, that integrates Predix with Amazon Alexa Voice Service (AVS) (<https://developer.amazon.com/alexa>, <https://developer.amazon.com/public/solutions/alexa/alexa-voice-service/content/avs-api-overview>). AVS allows developers to add smart voice-related functionality to their products having a microphone and speaker. You will use its capabilities to create a simple survey App.



- Backend.** Based on NodeJS and ExpressJS, it should be hosted in Predix cloud (locally for development). It will communicate with AVS to handle voice-related functionality and with **Predix Asset Service** to get survey questions and to store user answers. It will also serve a simple frontend. It should follow our usual best practices (well-documented, Airbnb linter to control the code quality, configuration, etc.).

When the visitor starts survey in the frontend, the backend will take a question from the survey stored in assets service, send it to AVS to synthesize the speech, and then proxy the AVS

REVIEW STYLE:

Final Review:
Community Review Board 1

Approval:
User Sign-Off 1

CHALLENGE LINKS:

[Review Scorecard](#)

CHALLENGE TERMS:

Standard Terms for TopCoder Competitions v2.1

GET THE UML TOOL:

- GetUML source code repository
- Mac disk image
- Java installer

SHARE:



TA3

When the visitor starts survey in the frontend, the backend will take a question from the survey stored in assets service, send it to AVS to synthesize the speech, and then proxy the AVS response to play the resulting speech at the frontend side. Once the visitor has listened the question, he will answer to it, using his voice and microphone. User's voice input will be proxied by the backend into AVS to recognize the answer. Once the answer is recognized, the backend will store the answer into Predix Asset, and will go to the next question, until all questions are completed.

2. **Frontend.** We need a very simple frontend for this challenge, you may use any framework you feel comfortable with, but be sure you should adhere the best practices for that framework. The frontend should provide very simple controls to start the survey, complete an answer to a question, to do any related operations; and visual aids to facilitate understanding, when user is supposed to listen a question, and when he is supposed to answer. Obviously, the frontend should utilize the microphone and speakers.

3. **AVS and Predix Assets.** These are provided by Amazon and Predix platforms. Be sure to document in details how to set up and configure them to work smoothly with your app.

Assets service should hold both the surveys and replies to the survey. For a survey we need to store: title, uri (by which the survey will be actually referenced in all APIs), and a set of questions, each with its own id. You should provide a convenient way to load such surveys to the assets service, and provide a couple of sample surveys with 5-10 questions each.

A single user response to a survey should hold uri of the response, user name, uri of the survey that was answered, and a set of responses. Each answer, beside the answer itself, should hold a reference to the answered question (think from the perspective that if we are to implement editing of the surveys, and a survey is changed at some point, we want to be sure, that the previous survey responses still can be interpreted correctly, and holding references to the answered question helps here).

In general, consider it as a simple proof of concept application. No need for fancy / complex features beside the main functional idea described above, but the code and the resulting app appearance should still look and feel good. Should you have any doubts, don't hesitate to ask in the challenge forums!

Final Submission Guidelines

Submit your code along with verification instructions and demo video.

Payments

Topcoder will compensate members in accordance with the our standard payment policies, unless otherwise specified in this challenge. For information on payment policies, setting up your profile to receive payments, and general payment questions, please refer to <https://help.topcoder.com/hc/en-us/articles/217482038-Payment-Policies-and-Instructions>.

Reliability Rating and Bonus

For challenges that have a reliability bonus, the bonus depends on the reliability rating at the moment of registration for that project. A participant with no previous projects is considered to have no reliability rating, and therefore gets no bonus. Reliability bonus does not apply to Digital Run winnings. Since reliability rating is based on the past 15 projects, it can only have 15 discrete values. [Read more](#).

TA4

IBM Cognitive ChatBot - Misc Updates II

[Frontend](#)
[Angular 2+](#)
[Cognitive](#)
[Express](#)
[IBM Bluemix](#)
[IBM Cognitive](#)
[IBM Watson](#)
[JavaScript](#)
[Node.js](#)
[NodeJS](#)

URL: <https://www.topcoder.com/challenge-details/30058908/?type=develop>

PRIZES

Entrega: 10 dias
(realizado em 2 dias)

1st
\$300

Unregister

Submit

The challenge is finished. [Show Deadlines](#)

[DETAILS](#)
[REGISTRANTS \(23\)](#)
[SUBMISSIONS \(6\)](#)
[CHALLENGE FORUM](#)

Challenge Overview

This F2F challenge is a part of **IBM Cognitive ChatBot** series, where we develop a chat bot based on **IBM Cognitive Conversation Service**, along with **Angular 4** frontend and **Express.js / Node.js** backend (ask in the challenge forum for authorization to access these repositories; conversation model for Cognitive service can be found in `/docs/topcoder_chatbot.json` in the backend repo). In this challenge you will do the following updates of the existing code:

- Right now all communications are treated by the frontend as purely textual. You should see however, that some conversation nodes assume `get_option` and `get_multiple_options` actions. You will implement the proper support for these. In the first case, the frontend should include into the question bubble buttons with valid options, assuming that user should click one of them to answer the question. In the second case, the frontend should include into the question bubble the list of options with checkboxes, and submit button, which sends the answer.
- At the welcome screen, right now clicking on the avatar, if all other data are introduced, immediately redirects to the chat. This should be updated, so that clicking on the avatar just selects (showing the same green border, visible when hovered), and activates Continue button. Only explicit click on the Continue button should take to the chat (if all other data are filled correctly). Also, please comment out the Plus Sign button, assumed to add custom avatars.
- When ChatBot does not understand something, it replies with *I am just an example of what you can build with IBM Watson and Topcoder's Cognitive Community. If you would like to learn more, go here to launch live chat with Topcoder*, where [here](#) is a link. Please (1) update the link's style to be underlined; (2) change the reference to point to www.topcoder.com.
- Right now in the frontend the backend URL is configured as a constant inside `/src/app/chat/chat.component.ts`. Please, create a proper configuration setup and move this url to config file. (e.g., look how it is done in the backend, where `node config` is used to handle configuration).

Final Submission Guidelines

Submit Git patches for the repositories, and a link to your solution deployed to IBM Bluemix:

Payments

Topcoder will compensate members in accordance with the our standard payment policies, unless otherwise specified in this challenge. For information on payment policies, setting up your profile to receive payments, and general payment questions, please refer to <https://help.topcoder.com/hc/en-us/articles/217482038-Payment-Policies-and-Instructions>.

Reliability Rating and Bonus

For challenges that have a reliability bonus, the bonus depends on the reliability rating at the moment of registration for that project. A participant with no previous projects is considered to have no reliability rating, and therefore gets no bonus. Reliability bonus does not apply to Digital Run winnings. Since reliability rating is based on the past 15 projects, it can only have 15 discrete values. [Read more](#).

REVIEW STYLE

Final Review:
Community Review Board 3

Approval:
User Sign-Off 3

CHALLENGE LINKS:
[Review Scorecard](#)

CHALLENGE TERMS:
[Standard Terms for TopCoder Competitions v2.1](#)

GET THE UML TOOL:
[Github source code repository](#)
[Mac disk image](#)
[Java installer](#)

SHARE:



TA5

IBM Cognitive ChatBot - Misc Updates

[Frontend](#)
[Angular 2+](#)
[Cognitive](#)
[Express](#)
[IBM Bluemix](#)
[IBM Cognitive](#)
[IBM Watson](#)
[Node.js](#)
[NodeJS](#)

URL: <https://www.topcoder.com/challenge-details/30058879/?type=develop>

PRIZES

Entrega: 10 dias
(realizado em 2 dias)

Prize: \$200

The challenge is finished

Unregister Submit

Show Deadlines

[DETAILS](#)
[REGISTRANTS \(36\)](#)
[SUBMISSIONS \(4\)](#)
[CHALLENGE FORUM](#)

Challenge Overview

This F2F challenge is a part of **IBM Cognitive ChatBot** series, where we develop a chat bot based on **IBM Cognitive Conversation Service**, along with **Angular 4 frontend** and **ExpressJS / NodeJS backend** (ask in the challenge forum for authorization of access to these repositories). In this challenge you will do the following updates to the existing code:

- Replace everywhere the IBM Bee logo with Topcoder's hat logo (i.e. the one without Topcoder name [get the source here](#));
- Rename the bot to Topcoder Chatbot (everywhere - UI, documentation, etc.);
- Ensure that user avatar selected on the login screen is then actually used inside the chat;
- Right now in all questions allowing to answer with a list of entries, the user is required to type different entries in different messages, and the answer to the question is terminated by saying **done**. All around such mechanic should be changed so that user replies by a single message, containing comma, or otherwise separated list on entries. E.g. in this dialog:



The user should be able to reply *director*, *manager* or *director and managers*.

- Hide the attachment, picture and microphone icons in the chat input line.
- When the bot does not understand something it says *I am sorry, I don't understand*. Change this with *I am just an example of what you can build with IBM Watson and Topcoder's Cognitive Community*. If you would like to learn more, [go here to launch live chat with Topcoder](#). The word *here* in the last sentence will be a [link](#) somewhere. We'll insert the actual link later, but please ensure that it is correctly styled as a link and clickable.

Final Submission Guidelines

Submit git patches for the repositories.

Payments

Topcoder will compensate members in accordance with our standard payment policies, unless otherwise specified in this challenge. For information on payment policies, setting up your profile to receive payments, and general payment questions, please refer to <https://help.topcoder.com/hc/en-us/articles/217482038-Payment-Policies-and-Instructions>.

Reliability Rating and Bonus

For challenges that have a reliability bonus, the bonus depends on the reliability rating at the moment of registration for that project. A participant with no previous projects is considered to have no reliability rating, and therefore gets no bonus. Reliability bonus does not apply to Digital Run winnings. Since reliability rating is based on the past 15 projects, it can only have 15 discrete values. [Read more](#).

REVIEW STYLE

Final Review
Community Review Board 1

Approval
User Sign-Off 1

CHALLENGE LINKS
Review Scorecard

CHALLENGE TERMS
1 Standard Terms for TopCoder Competitions v2.1

GET THE UML TOOL

- Github source code repository
- Mac disk image
- Java installer

SHARE



TA6

Authentication Solution Angular4 and Material UI Prototype Challenge

UI Prototype Competition | TCODB | Angular 2+ | CSS | HTML5 | Heroku | JSON | JavaScript | HTML | URL: <https://www.topcoder.com/challenge-details/30058867/?type=develop>

PRIZES

Entrega: 7 dias

1st \$1,400

2nd \$700

RELIABILITY BONUS: \$420

The challenge is finished. [Show Deadlines](#)

[DETAILS](#) | [REGISTRANTS \(49\)](#) | [SUBMISSIONS \(4\)](#) | [CHALLENGE FORUM](#)

Challenge Overview

Welcome to **Authentication Solution Angular4 Material UI Prototype Challenge**

The main task of this challenge is need to create the Authentication Solution Application by using [Angular4 & Material UI components](#), by following the provided storyboard

For this challenge you need focus on Desktop screen size only.
Let's discuss any questions you have on challenge forum.

Final Submission Guidelines

Important:

- You must use [Angular4](#) & [Material UI](#) components as base framework for this submission challenge
- Focus on Desktop screen size only
- Storyboard created by follow [Google Material Design](#) style.
- For this challenge as possible, use standard styling & function based on [Material UI components](#)
- Explore <https://material.angular.io/components> to get familiar with the components needed for this challenge
- For this prototype submission we need you to build using [Angular4](#) with [TypeScript](#), [HTML5](#), [CSS3](#) and [JSON](#) as main technologies.
- You can choose some [Angular4](#) starter seed to help you build the UI Prototype.
- Use [SCSS/SASS](#) for the submission styling
- [Angular](#) Components must re-usable. Component should be derived based on the configuration. Configuration file to contain the required configurations required to render the page
- Use [Lintes](#), [Jest](#) and other best practice library to maintain your code quality.
- Your submission must follow [Angular Coding guidelines](#): <https://angular.io/guide/styleguide>
- Put clear comment/documentation for all [Angular](#) components.
- All table datas need loaded by [JSON](#).
- Create [HTTP](#) Services to pull all data from [JSON](#).
- Create Routes that use proper naming with existing pages
- Create [README](#) md file that explains your UI Prototype submission structure and commands
- Create clear documentation to set up your submission locally and on [Heroku](#).
- Also, you must put your submission on [Heroku](#) for client review. Include the link/url in your submission

General Requirements

- Check this url for the Prototype clickflow: <https://marvelapp.com/9aghcjaj/screen/31764057>
- Design Guidelines: <https://marvelapp.com/9aghcjaj/screen/31364308>
- Storyboard Design: [Storyboard.zip](#)
- Use minimum width of 1280px for the page layout.
- Based on the storyboard, do not build the header that show the browser toolbar
- Click Sign out menu from left sidebar Need load the modal window and then take user to homepage.

Page Requirements

1). Login

- On the left side, text need editable
- Put login form on the right side
- When user focus on the inputs need show up the label on top. Refer to this example: <https://material.angular.io/components/input/examples>
- Forgot Password & Sign up can be dead links

DOWNLOADS

- [Storyboard Design Challenge](#)

ELIGIBLE EVENTS

- 2018 TopCoder(70) Open

REVIEW STYLE

- Final Review: [Community Review Board](#) 3

Approval

- User Sign-Off 3

CHALLENGE LINKS

- [Screening Scorecard](#)
- [Review Scorecard](#)

CHALLENGE TERMS

- [Standard Terms for TopCoder Competitions v2.1](#)
- [Approve NDA v2.0](#)

GET THE UML TOOL

- [Github](#) source code repository
- [Mac](#) disk image
- [Java](#) installer

SHARE:



TA6

- Forgot Password & Sign up can be dead links
- Sign In button take user to Admin page

2). Admin

- On top, display header bar with the application title
- Hamburger menu can be dead link
- On the right side of header bar, you need show badge of username
- On the left sidebar, click those menu will take to different pages
- Admin page need display these following box:
 - User Administration
 - LDAP Configuration
 - Activity Log
- Click Open button take user to related pages.

3). User Administration

- Breadcrumb displayed on top. Click Admin link need take to previous Admin page.
- Page consists of 2 box, on top user will see the Username input & Roles dropdown
- Display the label when user focus on the input
- Follow this styling for the dropdown: <https://material.angular.io/components/select/examples>
- Actions buttons: Search OR add button
- By default display some rows on the bottom
- Follow this styling for slide button: <https://material.angular.io/components/slide-toggle/examples>
- Put dead link for Edit & Delete button
- If user click Search, need find data with related Username & roles
- If user click Search, need add new data to row
- Done button take user back to Admin page

4). LDAP Configuration

- Config name input placed on top box
- Display current Config on table rows
- On the right side, delete button shows up
- User need able to scroll down, you can use standard scrollbar
- On the bottom box, there's Configuration table input that allow user to filled the data
- Test Connection need load modal
- Save button take user to Admin page

5). Activity Log

- This will consist of all Activity Log
- Match data like provided storyboard screens

6). Policy

- Policies Configuration box shows up when click the Policy menu
- On top box, user will see the input for Policy name & Description
- Click Search OR Add button need show different flow
- Policy table row displayed on the bottom, match content like provided screenshot
- On the very bottom box, you can see Rules and Headers tab
- Example of Material UI tabs: <https://material.angular.io/components/tabs/overview>
- Click Add new need show up the Add/Edit Rules modal window
- Example of Material UI dialog: <https://material.angular.io/components/dialog/overview>
- Click Headers Tabs, there are some table like Headers table, Static & User Attribute tab
- Need able to let user Add/Remove the Header. Click the Add/Remove button take user to Header page
- Show added rules need display the table on the bottom
- Cancel & Apply button take to Policy page

7). Header

- HTTP Header Details show shows up when user click the Header button
- Open button take user to HTTP Headers Details page
- On top of page, user will see the multi-select input
- Need able to move the selected row to the left or right side
- All headers need loaded by default
- User also need able to search the policies
- Click Add Policy button take user to Policy page
- On Create/Modify Header, user need filled Name & Description
- Then, click Add button need load Static or User Attribute on the bottom
- Done button take user to the Header page

8). Preview

- User need filled the User ID (NT ID), then click Preview button need load data like provided storyboard
- Need able to scrolling down the data
- Done button take user to the Preview page

TA6

Done button take user to the Preview page

CODE REQUIREMENTS:

HTML5

- Provide comments on the page elements to give clear explanation of the code usage. The goal is to help future developers understand the code.
- Please use clean INDENTATION for all HTML code so future developers can follow the code.
- All HTML code naming should not have any conflicts or errors.

CSS3

- Provide comments on the SCSS code. We need comments to give a clear explanation of the code usage. The goal is to help future developers understand the code.
- Please use clean INDENTATION for all code so developers can follow.
- All SCSS naming should not have any conflicts.
- As possible use CSS3 style when create all styling.

Javascript

- You must using Typescript (.ts) file format
- Use Linter, Jest and other best practice library to maintain your code quality.
- Your submission must follow Angular Coding guidelines: <https://angular.io/guide/styleguide>
- Put clear comment/documentation for all Angular components.
- All JavaScript must not have a copyright by a third party.
- It is fine to use GPL/MIT/Open Source code.
- You are encouraged to use your own scripts, or scripts that are free, publicly available and do not have copyright statements or author recognition requirements anywhere in the code.

IMAGES/SVG

- Images should be properly compressed while still having good visual quality
- Please use best practice repetition usage of background based image
- Please use sprites when using icons for your submission.
- If any icons are required, ask and they will be provided

Web Browsers Requirements

- IE11
- Microsoft Edge
- Firefox latest version
- Safari latest version
- Chrome latest version

Payments

Topcoder will compensate members in accordance with the our standard payment policies, unless otherwise specified in this challenge. For information on payment policies, setting up your profile to receive payments, and general payment questions, please refer to <https://help.topcoder.com/hc/en-us/articles/217482038-Payment-Policies-and-Instructions>.

Reliability Rating and Bonus

For challenges that have a reliability bonus, the bonus depends on the reliability rating at the moment of registration for that project. A participant with no previous projects is considered to have no reliability rating, and therefore gets no bonus. Reliability bonus does not apply to Digital Run winnings. Since reliability rating is based on the past 15 projects, it can only have 15 discrete values. [Read more.](#)

APÊNDICE I – ESTUDO DE CASO: SUGESTÕES DE MELHORIAS CICLO 1 E CICLO 2

CICLO 1		
Participante	Declaração	Mapeamento Modelo
P1	A informação do enunciado da tarefa esta adequada para ter uma noção geral da necessidade. Contudo, a documentação complementar poderia ter trazido mais detalhes do contexto da empresa (conforme mencionado nos itens anteriores). Além de um detalhamento de quem seria o público alvo que iria receber o relatório entregue como parte da atividade.	<ul style="list-style-type: none"> • Documentação Complementar
P3	Poderia ser disponibilizada em PDF , para evitar problemas de compatibilidade. Um sumário da tarefa na própria interface do TopCoder ajudaria, pois é necessária muita boa vontade para entender o que a tarefa propõe com apenas o título e as informações da página inicial do TopCoder.	<ul style="list-style-type: none"> • Resumo da Tarefa
P4	Descreverem melhor os requisitos e a forma em que será feita a avaliação e escolha, assim como o teste da qualidade da solução.	<ul style="list-style-type: none"> • Detalhamento da Tarefa • Critérios de Teste e Aceitação
P7	A tarefa tinha mockups e a descrição do comportamento, acho que, de repente, poderia ajudar um vídeo ou algo assim de um dos stakeholders comentando sobre o contexto e a necessidade daquela implementação.	<ul style="list-style-type: none"> • Links Relacionados • Documentação Complementar
P9	A documentação inicial foi bem fraca e faltaram muitas informações. Então penso que a documentação inicial já poderia ter sido melhor desenvolvida. Uma pessoa postou no forum várias perguntas a respeito da tarefa, que foram de grande ajuda para realizar a mesma, porém algumas respostas não foram satisfatórias e algumas duvidas ainda permaneceram.	<ul style="list-style-type: none"> • Resumo da Tarefa • Detalhamento da Tarefa
P10	Nesta atividade, pouco poderia ser feito. Acredito que um checklist para indicar que a tarefa foi terminada corretamente poderia auxiliar.	<ul style="list-style-type: none"> • Critérios de Teste e Aceitação

CICLO 1		
Participante	Declaração	Mapeamento Modelo
P11	Acredito que poderiam haver vídeos explicativos ou mesmo diagramas para melhorar a compreensão . Creio que elementos gráficos e audiovisuais facilitam a compreensão .	<ul style="list-style-type: none"> • Links Relacionados • Documentação Complementar
P12	Poderia ter um melhor detalhamento de telas e até de arquitetura de código conforme a tarefa necessitar.	<ul style="list-style-type: none"> • Detalhamento da Tarefa
P13	Não sei se é interessante para o TopCoder. Não parece ser o foco deles. Mas, se a documentação viesse com um pouco mais de detalhe , levando em consideração participantes iniciantes, provavelmente a quantidade de acertos na conclusão da tarefa seria maior.	<ul style="list-style-type: none"> • Detalhamento da Tarefa
P15	Acho que faltou mais clareza na forma de submissão das tarefas . A tarefa em si estava suficientemente clara, mas as questões que envolviam a própria plataforma poderiam ter sido melhor explicadas .	<ul style="list-style-type: none"> • Regras de Submissão da Tarefa
P16	A documentação deve estar em anexo já no momento da escolha da tarefa ao invés de termos de enviar um email solicitando a mesma. Acredito que assim facilitaria a escolha da tarefa.	<ul style="list-style-type: none"> • Links Relacionados • Documentação Complementar
P17	A questão do tempo de desenvolvimento, quando existe mais de uma entrega, para que possamos entender até quando e como funcionam esses "deadlines". Como a atividade era de design, faltou dizer se podia ou não utilizar o logo da competição (que no caso, não podia), mas eu usei e fiz a primeira entrega de um produto errado. E por fim, acho que faltou explicar o que era necessário conter nos arquivos entregáveis , já que era necessário explicar sobre o design do produto.	<ul style="list-style-type: none"> • Detalhamento da Tarefa • Critérios de Teste e Aceitação • Regras de Submissão da Tarefa

CICLO 1		
Participante	Declaração	Mapeamento Modelo
P20	A descrição tem que ser o mais precisa possível, acho que é o preço que se paga pela "impessoalidade". A documentação do front-end da aplicação era muito fraca , sendo esse front-end composto de inúmeros frameworks e ferramentas . Atualmente, devido a quantidade e variedade enorme de frameworks para front-end, é difícil que desenvolvedores full-stack estejam atualizados, mesmo dentro das empresas (tanto que existe vagas hoje só para front-end). Por isso a documentação do front-end deveria ser mais completa que a do back-end (pelo menos para um sistema em .Net). A do back-end, na minha opinião, estava razoável. No entanto eu possuo em torno de 8 anos de experiência com desenvolvimento .Net. Uma pessoa com menos experiência provavelmente teria tido dificuldades com o back-end também.	<ul style="list-style-type: none"> • Detalhamento da Tarefa • Ferramentas
P21	Poderiam deixar mais claras as opções de ferramentas necessárias para a execução por exemplo.	<ul style="list-style-type: none"> • Ferramentas

CICLO 2		
Participante	Declaração	Mapeamento Modelo
P1	Mais clareza e organização na descrição da tarefa . Principalmente porque ela continha informações relacionadas aos processos do TopCoder então fiquei confusa se todo o conteúdo era relacionado ao que eu precisava utilizar para realização da tarefa ou não. Precisei ler várias vezes (acho que 6 ou 7 vezes ao todo) toda a descrição da tarefa para compreender o que estava sendo solicitado.	<ul style="list-style-type: none"> • Detalhamento da Tarefa
P3	Para mim a maior dificuldade desta tarefa (e de outras) é identificar o que está sendo solicitado como entrega . Em vários locais não é informado o que é desejado (se um arquivo de um programa específico, como Sketch, Photoshop, Microsoft SQL) ou se é uma imagem genérica, tal como JPG, PNG, etc. Então ter uma caixa que indicasse claramente quais são os artefatos a serem entregues (e que os artefatos não fossem um parágrafo, mas sim uma seleção de itens pré-definidos) ajudaria não só a identificar o que é desejado mas também a filtrar as tarefas as quais me considero apto a realizar.	<ul style="list-style-type: none"> • Detalhamento da Tarefa • Critérios de Teste e Aceitação • Regras de Submissão da Tarefa

CICLO 2		
Participante	Declaração	Mapeamento Modelo
P4	<p>Presumo ser mais adequado solicitar a tarefa em forma de procedimento, em ordem, quase como um algoritmo: Passo 1 ... Passo 2 ... e assim por diante.</p>	<ul style="list-style-type: none"> • Detalhamento da Tarefa
P5	<p>O projeto tinha o escopo muito livre, não descrevendo na documentação o layout básico, visto que era uma complementação para um sistema já existente. Não continha informação do tráfego do site, nada para informar o total de acessos que pretendia ter por dia. Sendo uma extensão para uma aplicação já existente, não continha um layout prévio da aplicação a qual seria estendida, não disponibilizando informações de menu, tamanho de imagens, posicionamento, cores, etc.</p>	<ul style="list-style-type: none"> • Detalhamento da Tarefa
P6	<p>A tarefa em questão se utilizou de um linguajar extremamente vago, em que muitas coisas abstratas eram pedidas, do tipo, tem que passar a sensação de modernidade, agilidade, ter um significado escondido. Falava muito pouco da empresa em si, ou dos valores da mesma. Talvez algumas fotos mostrando onde a marca seria aplicada ajudassem a criar algo. Em geral, me pareceu uma conversa inicial de designer, que necessitaria de várias interações com o cliente para chegar a algo concreto, algo que a plataforma não tem suporte adequado.</p>	<ul style="list-style-type: none"> • Detalhamento da Tarefa
P7	<p>Nem todos repositórios de código tem um documento de README com instruções para setup do ambiente, o que às vezes dificulta iniciar o desafio. Uma tarefa que peguei anteriormente tinha screenshots do que era desejado, o que facilitou bastante a compreensão. A tarefa estar auto contida também ajuda, pois com documentação externa às vezes aparece, por exemplo, imagens de toda aplicação e você só tem que realizar uma tarefa dentro do sistema, aí tem que ficar encontrando quais assets são relacionados a tarefa.</p>	<ul style="list-style-type: none"> • Configuração do Ambiente

CICLO 2		
Participante	Declaração	Mapeamento Modelo
P8	No caso da tarefa que realizei havia um problema na documentação na plataforma do TopCoder. Mais especificamente faltou a documentação que indicava a necessidade de fazer o Download da carteira da Ethereum.	<ul style="list-style-type: none"> • Detalhamento da Tarefa • Links Relacionados • Documentação Complementar
P9	Diferentemente da tarefa 1 que realizei, essa tarefa estava bem documentada. Desta forma, achei a documentação satisfatória e não achei necessidade de pontos a melhorar na documentação. Talvez uma melhor descrição dos arquivos para entrega. E o feedback do checkpoint poderia ter sido mais rápido e detalhado.	<ul style="list-style-type: none"> • Critérios de Teste e Aceitação
P10	Melhorar a documentação, pensando em usuários mais experientes e menos experientes.	<ul style="list-style-type: none"> • Detalhamento da Tarefa
P11	Documentação complementar para auxiliar na montagem do ambiente (IDE, bibliotecas, etc).	<ul style="list-style-type: none"> • Configuração do Ambiente
P12	Pode ser melhor descritos os termos de aceite para a tarefa com os principais pontos que devem ser atendidos.	<ul style="list-style-type: none"> • Critérios de Teste e Aceitação
P13	Melhorar os detalhes técnicos do setup de ambiente (talvez criar uma sessão separada para isso). Especificar melhor as tarefas a serem entregues (especificar com mais clareza o que deve ser entregue).	<ul style="list-style-type: none"> • Configuração do Ambiente • Critérios de Teste e Aceitação
P14	Nesta tarefa em específico não vi nada, porém em outras tarefas achei que faltavam informações, documentação e que os links para os repositórios estavam quebrados. Cada pessoa que cadastra uma tarefa tem a sua percepção do que deve ser colocado no enunciado ou não. O que deixa as tarefas, talvez fora de um standard.	<ul style="list-style-type: none"> • Padronização
P15	Faltou explicar melhor sobre o setup do ambiente necessário. Só falava sobre a tarefa em sim e pouco sobre os requisitos para se iniciar a tarefa como: instalação do software e entendimento da teoria necessária de blockchain.	<ul style="list-style-type: none"> • Configuração do Ambiente • Detalhamento da Tarefa

CICLO 2		
Participante	Declaração	Mapeamento Modelo
P16	Tive de pedir ao copilot da tarefa a documentação necessária para tentar entender a mesma. Pois somente a descrição da tarefa não foi suficiente para entender o que deveria ser feito.	<ul style="list-style-type: none"> • Detalhamento da Tarefa
P17	Detalhar mais sobre essa ferramenta que necessitava de redesign fornecendo exemplos de projetos que o software modela, e suas aplicabilidades também.	<ul style="list-style-type: none"> • Ferramentas • Documentação Complementar
P18	Não ficou claro se certos passos deveriam ser descobertos por nós (desenvolvedores) ou se realmente foi "falha" da documentação. Alguns passos não eram 100% intuitivos, conforme mencionei no diário. Apesar de eu não ter pedido ajuda no blog da plataforma, tive que descobrir sozinho como executar alguns passos (conforme descrito no diário).	<ul style="list-style-type: none"> • Detalhamento da Tarefa
P19	Padronização na documentação de todas as tarefas do TopCoder através de um checklist.	<ul style="list-style-type: none"> • Padronização
P20	Poderia detalhar melhor os casos de uso e contextualizar a utilização do sistema. Poderia ter uma descrição mais aprofundada da stack e das bibliotecas utilizadas na solução.	<ul style="list-style-type: none"> • Configuração do Ambiente • Detalhamento da Tarefa



Pontifícia Universidade Católica do Rio Grande do Sul
Pró-Reitoria de Graduação
Av. Ipiranga, 6681 - Prédio 1 - 3º. andar
Porto Alegre - RS - Brasil
Fone: (51) 3320-3500 - Fax: (51) 3339-1564
E-mail: prograd@pucrs.br
Site: www.pucrs.br